Recursosinformáticos

LINUX

Dominar la administración del sistema

3ª edición

Sébastien ROHAUT

eni ediciones

Presentación de Linux

1. Bienvenidos al mundo Unix **29**

1.1 Un nuevo mundo 29
1.2 Historia de los ordenadores 30
1.2.1 Complejidad de los ordenadores 30
1.2.2 La inteligencia 30
1.3 El sistema operativo 31
1.4 El sistema Unix, una breve historia 34
1.4.1 De MULTICS a UNIX 34
1.4.2 El lenguaje C 36
1.4.3 Las licencias y el advenimiento de BSD y System V 37
1.4.4 La guerra de los Unix 38
1.4.5 La estandarización 39
1.4.6 Unix es un estándar 39
1.4.7 Unix en los ordenadores personales 40

2. El software libre **41**

2.1 Los orígenes del software libre 41
2.2 El proyecto GNU y la FSF 42
2.3 El open source 44
2.4 GNU/Linux 44
2.4.1 Linus Torvalds 44
2.4.2 El accidente 45
2.4.3 La primera versión oficial 45
2.4.4 El éxito comunitario 46
2.4.5 Los años 1994-1997 46
2.4.6 A partir de 1998: la explosión 47
2.4.7 Hoy y mañana 47

3. ¿Qué hardware es compatible con Linux? **48**

3.1 La arquitectura 48
3.2 Compatibilidad del hardware 50

www.fullengineeringbook.net

4. Elegir una distribución **53**

4.1 Debian 53
4.2 Ubuntu 54
4.3 Red Hat y Fedora 55
4.4 Mandriva (ex-Mandrake) y Mageia 56
4.5 openSUSE 57
4.6 Las demás 58
4.7 Los LiveCD 59

5. Obtener ayuda **60**

5.1 La ayuda propia de los comandos 60
5.2 La ayuda interna del shell 60
5.3 El manual en línea 61
5.3.1 Acceso 61
5.3.2 Estructura de una página 61
5.3.3 Navegación 62
5.3.4 Las secciones 63
5.3.5 Buscar por correspondencia 64
5.4 Las páginas info 64
5.5 Buscar ayuda en Internet 65

Instalación de Linux y de los paquetes de software

1. Instalar una Debian **67**

1.1 Soporte de instalación 67
1.2 Boot en el soporte 68
1.3 Elección de los idiomas y países 69
1.4 Anfitriones y usuarios 70
1.5 Cuentas root y usuarios 71
1.6 Particionado de los discos 71
1.7 Instalación 74
1.8 Configuración de los paquetes 76
1.9 Fin de instalación y reinicio 77

www.fullengineeringbook.net

2. Instalación de Fedora **78**

2.1 Soporte de instalación 78
2.2 Arranque del soporte 79
2.3 Idioma de la instalación 80
2.4 Resumen de la instalación 81
2.5 Teclado 82
2.6 Destino de instalación 83
2.7 Configuración de red 86
2.8 Reloj 87
2.9 Selección de paquetes 88
2.10 Usuarios 89
2.11 Fin de la instalación 90

3. Red Hat Package Manager **91**

3.1 Noción de paquete 91
3.2 El gestor RPM 92
3.3 Instalación, actualización y eliminación 92
3.4 El caso del núcleo 93
3.5 Consultas RPM 94
3.6 Verificación de los paquetes 95
3.7 Las dependencias 96
3.8 Actualizaciones automatizadas 97

4. YUM **97**

4.1 Configuración de los repositorios 97
4.2 Utilización de los repositorios 98
4.2.1 Refrescar la caché 98
4.2.2 Listar los paquetes 99
4.2.3 Instalar paquetes 100
4.2.4 Actualizaciones 101
4.2.5 Buscar un paquete 101
4.2.6 Suprimir un paquete 102

5. Debian Package **102**

www.fullengineeringbook.net

5.1 dpkg: el gestor de paquetes Debian . . . 102
5.2 Instalación, actualización y supresión . . . 103
5.3 Consultas con dpkg . . . 105
5.3.1 Listar los paquetes . . . 105
5.3.2 Encontrar un paquete que contiene un archivo . . . 106
5.3.3 Listar el contenido de un paquete . . . 106
5.4 Convertir paquetes . . . 106
5.5 La herramienta dselect . . . 108

6. Gestor APT . . . 109

6.1 Fundamentos . . . 109
6.2 Los repositorios . . . 109
6.2.1 Configuración . . . 109
6.2.2 Actualización de la base de datos . . . 111
6.3 Actualización de la distribución . . . 112
6.4 Buscar e instalar un paquete individual . . . 114
6.5 Cliente gráfico . . . 115

7. Instalar desde las fuentes . . . 116

7.1 Obtener las fuentes . . . 116
7.2 Requisitos y dependencias . . . 116
7.3 Ejemplo de instalación . . . 117
7.4 Desinstalación . . . 123
7.5 Las bases del Makefile . . . 124
7.5.1 Bases . . . 124
7.5.2 Makefile intermedio . . . 125
7.5.3 Un poco más complejo . . . 126

8. Gestionar las librerías compartidas . . . 129

8.1 Fundamentos . . . 129
8.2 Lugar de almacenamiento . . . 129
8.3 ¿Qué librerías vinculadas hay? . . . 131
8.4 Configurar la caché del editor de vínculos . . . 132

www.fullengineeringbook.net

El shell y los comandos GNU

1. El shell bash **135**

1.1 Función 135
1.2 Bash: el shell por defecto 136
1.2.1 Un shell potente y libre 136
1.2.2 Línea de comandos 137
1.3 Utilizar el shell 138
1.3.1 La introducción de datos 138
1.3.2 Sintaxis general de los comandos 138
1.3.3 Primer ejemplo concreto con cal 139
1.3.4 Encadenar los comandos 140
1.3.5 Visualizar texto con echo 140
1.3.6 Comandos internos y externos 141
1.3.7 Algunos atajos útiles 142
1.4 El historial de comandos 142

2. La gestión de los archivos **143**

2.1 El sistema de archivos 143
2.2 Los diferentes tipos de archivos 144
2.2.1 Los archivos ordinarios o regulares 144
2.2.2 Los catálogos 145
2.2.3 Los archivos especiales 145
2.3 Nomenclatura de los archivos 146
2.4 Las rutas 146
2.4.1 Estructura y nombre de ruta 146
2.4.2 Directorio personal 147
2.4.3 Ruta relativa 147
2.4.4 La virgulilla 148
2.4.5 cd 148
2.5 Los comandos básicos 149
2.5.1 Listar los archivos y los directorios 149
2.5.2 Gestionar los archivos y los directorios 151
2.5.3 Comodines: carácter de sustitución 156
2.5.4 Cierre de caracteres 158

www.fullengineeringbook.net

3. Buscar archivos **158**

3.1 Consideraciones generales 158
3.2 Criterios de búsqueda 159
3.2.1 -name 159
3.2.2 -type 159
3.2.3 -user y -group 160
3.2.4 -size 160
3.2.5 -atime, -mtime y -ctime 161
3.2.6 -perm 162
3.2.7 -links y -inum 162
3.3 Comandos 163
3.3.1 -ls 163
3.3.2 -exec 163
3.3.3 -ok 164
3.4 Criterios AND / OR / NOT 164
3.5 Encontrar ejecutables 165
3.5.1 whereis 165
3.5.2 which 165
3.5.3 locate 166

4. El editor vi **167**

4.1 Presentación 167
4.2 Funcionamiento 167
4.3 Los comandos 168
4.3.1 La introducción de datos 168
4.3.2 Salir y guardar 168
4.3.3 Desplazamiento 169
4.3.4 La corrección 170
4.3.5 Búsqueda en el texto 170
4.3.6 Comandos de sustitución 171
4.3.7 Copiar - Pegar 171
4.3.8 Sustitución 172
4.3.9 Otros 172

www.fullengineeringbook.net

5. Redirecciones **173**

5.1 Fundamentos 173
5.2 De salida 174
5.3 En entrada 174
5.4 Documento en línea 175
5.5 Los canales estándares 175
5.6 Apertura de canales 176
5.7 Filtro: definición 176
5.8 Pipelines/tuberías 177

6. Filtros y herramientas **177**

6.1 Extracción de los nombres y rutas 177
6.2 Búsqueda de líneas 178
6.2.1 grep 178
6.2.2 egrep 178
6.2.3 fgrep 180
6.2.4 sed 180
6.3 Columnas y campos 181
6.3.1 Columnas 181
6.3.2 Campos 182
6.4 Recuento de líneas 183
6.5 Ordenación de líneas 184
6.6 Eliminación de las líneas repetidas 185
6.7 Unión de dos archivos 186
6.7.1 En los campos comunes 186
6.7.2 Línea a línea 186
6.8 División de un archivo en partes 187
6.8.1 Recortar 187
6.8.2 Reconstruir 188
6.9 Sustitución de caracteres 188
6.9.1 Lista de caracteres 188
6.9.2 Tabulaciones y espacios 190
6.10 Visualización de texto 190
6.10.1 En pantalla completa 190

www.fullengineeringbook.net

6.10.2 El principio de un archivo . . . 191
6.10.3 Fin y modo de espera de archivo . . . 191
6.10.4 Dar formato a una salida . . . 192
6.11 Duplicación del canal de salida estándar . . . 192
6.12 Comparación de archivos . . . 193
6.12.1 diff . . . 193
6.12.2 cmp . . . 194
6.13 Plazo de espera . . . 195
6.14 Controlar el flujo . . . 195

7. Los procesos . . . 196

7.1 Definición y entorno . . . 196
7.2 Estados de un proceso . . . 197
7.3 Ejecución en segundo plano . . . 197
7.4 Background, foreground, jobs . . . 198
7.5 Lista de los procesos . . . 199
7.6 Parada de un proceso/señales . . . 201
7.7 nohup . . . 202
7.8 nice y renice . . . 202
7.9 time . . . 203

8. Más todavía del bash . . . 203

8.1 Alias . . . 203
8.2 Agrupación de comandos . . . 204
8.3 Relación y ejecución condicional . . . 205

9. Las variables . . . 205

9.1 Nomenclatura . . . 206
9.2 Declaración y asignación . . . 206
9.3 Acceso y visualización . . . 206
9.4 Supresión y protección . . . 207
9.5 Export . . . 208
9.6 Llaves . . . 208
9.7 Llaves y sustitución condicional . . . 209
9.8 Variables de sistema . . . 210

www.fullengineeringbook.net

9.9 Variables especiales . . . 211
9.10 Longitud de una cadena . . . 211
9.11 Tablas y campos . . . 211
9.12 Variables tipadas . . . 212

10. Configuración del bash . . . 213

10.1 Archivos de configuración . . . 213
10.1.1 Shell de conexión . . . 213
10.1.2 Shell simple . . . 213
10.1.3 Modo Bourne shell . . . 214
10.1.4 Modo no interactivo . . . 214
10.2 Comandos set . . . 214

11. Programación shell . . . 215

11.1 Estructura y ejecución de un script . . . 215
11.2 Argumentos de un script . . . 216
11.2.1 Parámetros de posición . . . 216
11.2.2 Redefinición de los parámetros . . . 217
11.2.3 Reorganización de los parámetros . . . 217
11.2.4 Salida de script . . . 218
11.3 Entorno del proceso . . . 218
11.4 Sustitución de comando . . . 220
11.5 El programa test . . . 220
11.5.1 Pruebas en una cadena . . . 220
11.5.2 Pruebas sobre los valores numéricos . . . 221
11.5.3 Pruebas sobre todos los archivos . . . 222
11.5.4 Pruebas combinadas por criterios Y, O, NO . . . 223
11.5.5 Sintaxis ligera . . . 223
11.6 if ... then ... else . . . 223
11.7 Evaluación múltiple . . . 224
11.8 Introducción de cadena por el usuario . . . 226
11.9 Los bucles . . . 226
11.9.1 Bucle for . . . 227
11.9.2 Bucle while . . . 230
11.9.3 Bucle until . . . 231

www.fullengineeringbook.net

11.9.4 true y false . . . 231
11.9.5 break y continue . . . 232
11.9.6 Bucle select . . . 232
11.10 Las funciones . . . 233
11.11 Cálculos y expresiones . . . 234
11.11.1 expr . . . 234
11.11.2 Cálculos con bash . . . 236
11.12 Punteros . . . 236
11.13 Tratamiento de señales . . . 237
11.14 Comando ":" . . . 238

12. SQL . . . 238

12.1 Presentación . . . 238
12.2 Peticiones de selección . . . 239
12.2.1 Select . . . 239
12.2.2 Distinct . . . 239
12.2.3 Where . . . 240
12.3 Las expresiones y las funciones . . . 240
12.4 La cláusula ORDER BY . . . 241
12.5 La cláusula GROUP BY . . . 241
12.6 Las combinaciones . . . 242
12.7 Subconsultas . . . 243
12.8 Las inserciones . . . 243
12.9 Actualizaciones . . . 243
12.10 Eliminación . . . 244

Los discos y el sistema de archivos

1. Representación de los discos . . . 245

1.1 Nomenclatura . . . 245
1.1.1 IDE . . . 245
1.1.2 SCSI, SATA, USB, FIREWIRE, etc. . . . 246
1.2 Casos especiales . . . 247
1.2.1 Controladores específicos . . . 247

www.fullengineeringbook.net

1.2.2 Virtualización . . . 247
1.2.3 SAN, iSCSI, multipathing . . . 247

2. Operaciones de bajo nivel . . . 247

2.1 Información . . . 247
2.2 Modificación de los valores . . . 249

3. Elegir un sistema de archivos . . . 251

3.1 Fundamentos . . . 251
3.1.1 Definición de sistema de archivos . . . 251
3.1.2 Representación . . . 251
3.1.3 Los metadatos . . . 252
3.1.4 Los nombres de los archivos . . . 252
3.1.5 El archivo de registro . . . 252
3.2 Los sistemas de archivos en Linux . . . 253
3.2.1 ext2 . . . 253
3.2.2 ext3 . . . 253
3.2.3 ext4 . . . 254
3.2.4 btrfs . . . 254
3.2.5 reiserfs . . . 255
3.2.6 xfs . . . 255
3.2.7 vfat . . . 256

4. Particionado . . . 257

4.1 Particionado lógico . . . 257
4.2 Particionado MBR . . . 258
4.2.1 MBR y BIOS . . . 258
4.2.2 MBR . . . 258
4.2.3 Las particiones . . . 258
4.2.4 EBR . . . 260
4.2.5 PBR . . . 260
4.2.6 Tipos de particiones . . . 260
4.3 Particionado GPT . . . 261
4.3.1 GPT y UEFI . . . 261
4.3.2 GUID . . . 262

www.fullengineeringbook.net

4.3.3 LBA 0 . . . 262
4.3.4 LBA 1 . . . 263
4.3.5 LBA 2 a 33 . . . 263
4.3.6 Tipos de particiones . . . 264
4.4 Manejar las particiones . . . 265
4.4.1 Herramientas de gestión de particiones . . . 265
4.4.2 Manipular las particiones MBR . . . 266
4.4.3 Manipular las particiones GPT . . . 270

5. Manejar los sistemas de archivos . . . 271

5.1 Definición básica . . . 271
5.1.1 Bloque . . . 271
5.1.2 Superbloque . . . 272
5.1.3 Tabla de inodos . . . 272
5.1.4 Tablas de catálogo . . . 274
5.1.5 Hard link . . . 275
5.2 Crear un sistema de archivos . . . 275
5.2.1 mkfs, sintaxis general . . . 275
5.2.2 Un primer ejemplo en ext2 . . . 276
5.2.3 ext2, ext3 y ext4 . . . 277
5.2.4 reiserfs . . . 280
5.2.5 xfs . . . 281
5.2.6 btrfs . . . 281
5.2.7 vfat . . . 282

6. Acceder a los sistemas de archivos . . . 284

6.1 mount . . . 284
6.1.1 Montaje por periférico . . . 284
6.1.2 Opciones de montaje . . . 287
6.1.3 umount . . . 288
6.1.4 /etc/fstab . . . 289
6.1.5 El caso de los CD e imágenes ISO . . . 291

7. Controlar el sistema de archivos . . . 292

7.1 Estadísticas de ocupación . . . 292

www.fullengineeringbook.net

7.1.1 Por sistema de archivos . . . 292
7.1.2 Por estructura . . . 293
7.2 Comprobar, ajustar y arreglar . . . 294
7.2.1 fsck . . . 294
7.2.2 badblocks . . . 295
7.2.3 dumpe2fs . . . 296
7.2.4 tune2fs . . . 297

8. La swap . . . 299

8.1 ¿Porqué crear una swap? . . . 299
8.2 Tamaño óptimo . . . 299
8.3 Crear una partición de swap . . . 300
8.4 Activar y desactivar la swap . . . 300
8.4.1 Activación dinámica . . . 300
8.4.2 En /etc/fstab . . . 301
8.5 En caso de emergencia: archivo de swap . . . 301
8.6 Estado de la memoria . . . 302
8.6.1 free . . . 302
8.6.2 Memoria reservada . . . 303
8.6.3 meminfo . . . 304

9. Las cuotas de disco . . . 305

9.1 Definiciones . . . 305
9.2 Implementación . . . 306

10. Los permisos de acceso . . . 307

10.1 Los permisos básicos . . . 307
10.1.1 Permisos y usuarios . . . 307
10.1.2 Significado . . . 308
10.2 Modificación de los permisos . . . 309
10.2.1 Mediante símbolos . . . 309
10.2.2 Sistema octal . . . 310
10.3 Máscara de permisos . . . 311
10.3.1 Restringir permisos de manera automática . . . 311
10.3.2 Cálculo de máscara . . . 312

www.fullengineeringbook.net

10.4 Cambiar de propietario y de grupo . . . 312
10.5 Permisos de acceso extendidos . . . 313
10.5.1 SUID y SGID . . . 313
10.5.2 Real / efectivo . . . 314
10.5.3 Sticky bit . . . 314
10.5.4 Permisos y directorios . . . 315

Inicio de Linux, servicios, núcleo y periféricos

1. Proceso de inicio . . . 317

1.1 La BIOS y UEFI . . . 317
1.1.1 BIOS . . . 317
1.1.2 UEFI . . . 318
1.1.3 Ajustes básicos . . . 319
1.2 El gestor de arranque . . . 320
1.3 GRUB . . . 321
1.3.1 Configuración . . . 321
1.3.2 Instalación . . . 322
1.3.3 Arranque y edición . . . 322
1.4 GRUB2 . . . 323
1.4.1 GRUB2, el sustituto de GRUB . . . 323
1.4.2 Configuración . . . 323
1.4.3 Arranque y edición . . . 326
1.4.4 Caso de GPT y UEFI . . . 327
1.5 Inicialización del núcleo . . . 327

2. init System V . . . 328

2.1 Funciones . . . 328
2.2 Niveles de ejecución . . . 329
2.3 /etc/inittab . . . 330
2.4 Cambio de nivel . . . 332
2.5 Configuración del sistema básico . . . 332
2.6 Nivel de ejecución . . . 333
2.7 Gestión de los niveles y de los servicios . . . 334

www.fullengineeringbook.net

2.7.1 Servicios en init.d . . . 334
2.7.2 Control manual de los servicios . . . 336
2.7.3 Modificación de los niveles de ejecución . . . 337
2.8 Consolas virtuales . . . 339
2.9 Los logins . . . 340
2.10 Parada . . . 341

3. systemd . . . 342

3.1 Fundamentos . . . 342
3.2 Unidades objetivo y servicios . . . 343
3.3 Acciones . . . 345
3.4 Interfaz gráfica . . . 347

4. upstart . . . 348

4.1 Fundamentos . . . 348
4.2 Archivos . . . 349
4.3 Nivel por defecto . . . 350
4.4 Compatibilidad con System V . . . 350
4.5 Comandos de control . . . 350
4.6 Activación y desactivación de un servicio . . . 351

5. Consultar el registro del sistema . . . 352

5.1 dmesg . . . 352
5.2 /var/log/messages . . . 354
5.3 /var/log/syslog . . . 354

6. Servicios y módulos del núcleo . . . 355

6.1 Presentación . . . 355
6.2 uname . . . 356
6.3 Gestión de los módulos . . . 357
6.3.1 lsmod . . . 358
6.3.2 depmod . . . 359
6.3.3 modinfo . . . 359
6.3.4 insmod . . . 361
6.3.5 rmmod . . . 362

www.fullengineeringbook.net

6.3.6 modprobe . . . 362
6.3.7 modprobe.conf y modprobe.d . . . 363
6.4 Carga de los módulos al inicio . . . 364
6.4.1 initrd . . . 364
6.4.2 Red Hat: /etc/rc.modules . . . 366
6.4.3 openSUSE: /etc/sysconfig/kernel . . . 367
6.4.4 Debian y Ubuntu: /etc/modules . . . 367
6.5 Parámetros dinámicos . . . 367
6.5.1 /proc y /sys . . . 367
6.5.2 sysctl . . . 369

7. Compilar un núcleo . . . 370

7.1 Obtener las fuentes . . . 370
7.1.1 Fuentes oficiales . . . 370
7.1.2 Fuentes de la distribución . . . 371
7.2 Las herramientas necesarias . . . 372
7.3 Configuración . . . 372
7.3.1 El .config . . . 372
7.3.2 Recuperar la configuración del núcleo . . . 373
7.3.3 make oldconfig . . . 374
7.3.4 make menuconfig . . . 375
7.3.5 make xconfig . . . 376
7.3.6 Algunas opciones de optimización . . . 377
7.4 Compilación . . . 379
7.5 Instalación . . . 381
7.6 Test . . . 383
7.7 Otras opciones . . . 383

8. Los archivos de periféricos . . . 383

8.1 Introducción . . . 383
8.2 Archivos especiales . . . 384
8.3 Crear un archivo especial . . . 386
8.4 Conocer su hardware . . . 386
8.4.1 Bus PCI . . . 386
8.4.2 Bus USB . . . 388

www.fullengineeringbook.net

8.4.3 Recursos físicos . . . 389
8.4.4 Otras herramientas . . . 392
8.5 El soporte del USB y del hotplug . . . 395
8.5.1 Los módulos . . . 395
8.5.2 Carga . . . 396
8.5.3 hotplug, usbmgr . . . 397
8.5.4 udev . . . 397

Las tareas administrativas

1. Administración de los usuarios . . . 401

1.1 Fundamentos . . . 401
1.1.1 Identificación y autenticación . . . 401
1.1.2 Los usuarios . . . 401
1.1.3 Los grupos . . . 402
1.1.4 Las contraseñas . . . 403
1.2 Los archivos . . . 403
1.2.1 /etc/passwd . . . 403
1.2.2 /etc/group . . . 404
1.2.3 /etc/shadow . . . 405
1.2.4 /etc/gshadow . . . 406
1.3 Gestión de los usuarios . . . 406
1.3.1 Creación . . . 406
1.3.2 Seguridad de las contraseñas . . . 408
1.3.3 Modificación . . . 411
1.3.4 Eliminación . . . 412
1.4 Gestión de los grupos . . . 412
1.4.1 Creación . . . 412
1.4.2 Modificación . . . 412
1.4.3 Eliminación . . . 413
1.4.4 Contraseña . . . 413
1.5 Comandos adicionales . . . 414
1.5.1 Conversión de los archivos . . . 414
1.5.2 Verificar la coherencia . . . 415

www.fullengineeringbook.net

1.5.3 Comprobar las conexiones . . . 415
1.5.4 Acciones de usuario . . . 416
1.6 Configuración avanzada . . . 418
1.6.1 /etc/default/useradd . . . 419
1.6.2 /etc/default/passwd . . . 419
1.6.3 /etc/default/su . . . 420
1.6.4 /etc/login.defs . . . 420
1.7 Notificaciones al usuario . . . 423
1.7.1 /etc/issue . . . 423
1.7.2 /etc/issue.net . . . 423
1.7.3 /etc/motd . . . 423
1.8 El entorno de usuario . . . 423
1.8.1 /etc/skel . . . 423
1.8.2 Scripts de configuración . . . 424
1.8.3 Grupos privados y setgid . . . 424
1.9 Resumen general de PAM . . . 425

2. La impresión . . . 428

2.1 Principio . . . 428
2.2 System V . . . 429
2.3 BSD . . . 430
2.4 CUPS . . . 430
2.4.1 Presentación . . . 430
2.4.2 Añadir una impresora . . . 432

3. Automatización . . . 437

3.1 Con cron . . . 437
3.1.1 Presentación . . . 437
3.1.2 Formatos . . . 438
3.1.3 Ejemplos . . . 438
3.1.4 crontab sistema . . . 439
3.1.5 Control de acceso . . . 439
3.2 Con at . . . 440
3.2.1 Presentación . . . 440
3.2.2 Formatos . . . 440

www.fullengineeringbook.net

3.2.3 Control de las tareas . . . 441
3.2.4 Control de acceso . . . 442
3.3 Con anacron . . . 442

4. Los registros (logs) del sistema . . . 443

4.1 Fundamentos . . . 443
4.2 Los mensajes . . . 444
4.3 Configuración de syslog . . . 444
4.4 El caso de rsyslog . . . 446
4.5 Los registros . . . 447
4.6 Journald . . . 447

5. Archivado y backup . . . 448

5.1 Las herramientas de copia de seguridad . . . 448
5.1.1 Comandos, planes, scripts . . . 449
5.1.2 Otros comandos . . . 450
5.2 tar . . . 450
5.2.1 Archivadores . . . 450
5.2.2 Listar . . . 451
5.2.3 Restauración . . . 451
5.2.4 Otros parámetros . . . 452
5.3 cpio . . . 453
5.3.1 Archivar . . . 453
5.3.2 Listar . . . 454
5.3.3 Restaurar . . . 454
5.4 dd . . . 455

6. El reloj . . . 456

6.1 Conocer la hora . . . 456
6.1.1 fecha . . . 456
6.1.2 hwclock . . . 458
6.2 Modificar el reloj físico . . . 458
6.2.1 Mediante date . . . 458
6.2.2 Mediante hwclock . . . 459
6.3 NTP . . . 459

www.fullengineeringbook.net

6.3.1 Fundamentos . . . 459
6.3.2 Cliente ntp . . . 460

7. Los parámetros regionales . . . 460

7.1 i18n y l10n . . . 460
7.2 Configuraciones locales . . . 461
7.2.1 Herramientas de la distribución . . . 461
7.2.2 Variables de entorno . . . 462
7.2.3 Husos horarios . . . 464
7.3 Codificación de los caracteres . . . 464

La red

1. TCP/IP . . . 467

1.1 Fundamentos . . . 467
1.2 Direccionamiento . . . 468
1.2.1 Clases . . . 468
1.2.2 Subredes . . . 469
1.2.3 Encaminamiento . . . 470
1.2.4 IPv6 . . . 471
1.3 Configuración . . . 472
1.3.1 El caso de las distribuciones de tipo Red Hat/Fedora . . . 472
1.3.2 Máquinas de tipo Debian . . . 475
1.3.3 Encaminamiento . . . 476
1.4 Herramientas de red . . . 477
1.4.1 FTP . . . 477
1.4.2 Telnet . . . 479
1.4.3 Ping . . . 480
1.4.4 Traceroute . . . 481
1.4.5 Whois . . . 482
1.4.6 Netstat . . . 482
1.4.7 IPTraf . . . 485
1.5 Archivos generales . . . 486
1.5.1 /etc/resolv.conf . . . 486

www.fullengineeringbook.net

1.5.2 /etc/hosts y /etc/networks . . . 486
1.5.3 /etc/nsswitch.conf . . . 487
1.5.4 /etc/services . . . 487
1.5.5 /etc/protocols . . . 489

2. Servicios de red xinetd . . . 489

2.1 Presentación . . . 489
2.2 Configuración . . . 490
2.3 Inicio y parada de los servicios . . . 492

3. Conexión PPP . . . 492

3.1 Elección y configuración del módem . . . 492
3.1.1 El caso de los Winmodems . . . 492
3.1.2 Los archivos periféricos . . . 493
3.1.3 Ajustar el puerto serie . . . 493
3.1.4 Los comandos AT . . . 494
3.2 PPP . . . 494
3.3 Conexión mediante la consola . . . 495
3.3.1 Manualmente . . . 495
3.3.2 Por los archivos . . . 496
3.3.3 Conexión . . . 497

4. OpenSSH . . . 498

4.1 Presentación . . . 498
4.2 Configuración . . . 498
4.3 Utilización . . . 499
4.4 Claves y conexión automática . . . 499
4.4.1 Lado cliente . . . 499
4.4.2 Lado servidor . . . 500
4.4.3 Copia automática . . . 500
4.5 Passphrase y agente SSH . . . 501

5. Montar un servidor DHCP . . . 502

5.1 Presentación . . . 502
5.2 Arranque del servidor dhcpd . . . 502

www.fullengineeringbook.net

5.3 Información básica . . . 503
5.4 Cliente . . . 504

6. Servidor DNS . . . 504

6.1 Presentación . . . 504
6.2 Inicio . . . 505
6.3 Configuración de Bind . . . 506
6.3.1 Configuración general . . . 506
6.3.2 Sección global . . . 507
6.3.3 Sección de zonas . . . 507
6.3.4 Zona de resolución . . . 508
6.3.5 Zona de resolución inversa . . . 508
6.3.6 Ejemplo . . . 509
6.3.7 Zonas especiales . . . 509
6.4 Archivos de zonas . . . 510
6.4.1 Definiciones . . . 510
6.4.2 Zona . . . 511
6.4.3 Zona de resolución inversa . . . 513
6.5 Diagnóstico de los problemas de configuración . . . 514
6.6 Consulta dig y host . . . 514

7. Correo electrónico . . . 517

7.1 Fundamentos . . . 517
7.2 postfix . . . 518
7.2.1 Configuración sencilla . . . 518
7.2.2 Alias de usuarios . . . 519
7.2.3 Prueba . . . 519
7.3 POP e IMAP . . . 520

8. Servicio HTTP Apache . . . 521

8.1 Presentación . . . 521
8.2 Parada/Reinicio . . . 521
8.3 Configuración . . . 521
8.4 Directivas generales . . . 522
8.5 Gestión del rendimiento . . . 523

www.fullengineeringbook.net

8.6 Los directorios, alias y ubicaciones . . . 523
8.6.1 Directory . . . 523
8.6.2 Alias . . . 524
8.7 Hosts virtuales . . . 525

9. Archivos compartidos . . . 526

9.1 NFS . . . 526
9.1.1 Inicio . . . 526
9.1.2 Compartir por el lado servidor . . . 526
9.1.3 Montaje por el lado cliente . . . 528
9.2 FTP . . . 529

10. Compartir Windows con Samba . . . 530

10.1 Presentación . . . 530
10.2 Configuración . . . 531
10.3 Archivos compartidos . . . 531
10.4 Impresoras compartidas . . . 532
10.5 Métodos de autenticación . . . 533
10.6 Correspondencia de los nombres y contraseñas . . . 533
10.7 Clientes SAMBA . . . 534
10.7.1 En línea . . . 534
10.7.2 Montaje . . . 534

La seguridad

1. Bases de seguridad . . . 535

1.1 Seguridad informática . . . 535
1.2 Controlar los privilegios especiales . . . 536
1.3 Comprobar los paquetes . . . 538
1.4 Política de la contraseñas . . . 538
1.5 Prohibir las conexiones . . . 540
1.5.1 /bin/false . . . 540
1.5.2 /etc/nologin . . . 541
1.5.3 /etc/securetty . . . 541

www.fullengineeringbook.net

1.6 Probar las contraseñas . . . 542
1.7 Buscar rootkits . . . 543
1.7.1 Fundamentos del rootkit . . . 543
1.7.2 chkrootkit . . . 544
1.8 Los virus . . . 545
1.9 Los límites del usuario . . . 548
1.10 Los derechos SUDO . . . 549
1.10.1 Proporcionar privilegios extendidos . . . 549
1.10.2 Sintaxis de /etc/sudoers . . . 550
1.11 Auditoría más completa . . . 552
1.12 Los boletines de seguridad . . . 552
1.12.1 CERT: Computer Emergency Response Team . . . 552
1.12.2 SecurityFocus . . . 554
1.12.3 Los boletines de las distribuciones . . . 555
1.12.4 Los parches correctores . . . 556

2. Seguridad de servicos y de red . . . 557

2.1 Comprobar los puertos abiertos . . . 557
2.1.1 Los sockets . . . 557
2.1.2 Información desde netstat . . . 557
2.1.3 La herramienta nmap . . . 559
2.2 Suprimir los servicios inútiles . . . 561
2.2.1 Cuestiones generales . . . 561
2.2.2 Servicios standalone . . . 562
2.2.3 Servicios xinetd . . . 562
2.3 Los tcp_wrappers . . . 562
2.4 Netfilter . . . 564
2.4.1 Presentación . . . 564
2.4.2 Vida de un paquete . . . 565
2.4.3 Principio de las reglas . . . 566
2.4.4 Destinos de reglas . . . 566
2.4.5 Primer ejemplo . . . 566
2.4.6 Operaciones básicas . . . 567
2.4.7 Criterios de correspondencia . . . 568
2.4.8 Tablas . . . 569
2.4.9 Guardar las configuraciones . . . 570

www.fullengineeringbook.net

2.5 UFW . . . 570
2.5.1 Activación y estado . . . 571
2.5.2 Reglas por defecto . . . 571
2.5.3 Gestión de reglas . . . 572
2.5.4 Limitaciones . . . 574
2.6 GPG . . . 574
2.6.1 Un clon de PGP . . . 574
2.6.2 Generar claves . . . 575
2.6.3 Generar una clave de revocación . . . 577
2.6.4 Gestionar el almacén de claves . . . 579
2.6.5 Exportar la clave pública . . . 579
2.6.6 Importar una clave . . . 580
2.6.7 Firmar una clave . . . 581
2.6.8 Firmar y cifrar . . . 584

X Window

1. ¿Cómo funciona un entorno gráfico? . . . 587

1.1 X Window System . . . 587
1.1.1 Un modelo cliente/servidor . . . 587
1.1.2 El gestor de ventanas . . . 589
1.1.3 Los widgets y los toolkits . . . 590
1.1.4 Los escritorios virtuales . . . 592
1.2 Los entornos de escritorio . . . 593

2. Xorg . . . 595

2.1 Presentación . . . 595
2.2 Instalación . . . 596
2.3 Configuración . . . 597
2.3.1 Por distribución . . . 597
2.3.2 Xorgcfg . . . 598
2.3.3 Xorgconfig . . . 599
2.3.4 X . . . 599
2.4 Estructura de xorg.conf . . . 599

www.fullengineeringbook.net

2.4.1 División . . . 599
2.4.2 Valores booleanos . . . 600
2.4.3 Sección InputDevice . . . 600
2.4.4 Sección Monitor . . . 602
2.4.5 Sección Modes . . . 603
2.4.6 Sección Device . . . 605
2.4.7 Sección Screen . . . 606
2.4.8 Sección ServerLayout . . . 607
2.4.9 Sección Files . . . 608
2.4.10 Sección Modules . . . 609
2.4.11 Sección ServerFlags . . . 610
2.4.12 Sección Extensions . . . 610
2.4.13 xorg.conf.d . . . 611
2.4.14 xorg.conf al completo . . . 611
2.5 Probar e iniciar X . . . 614
2.5.1 Comprobar la configuración . . . 614
2.5.2 El registro . . . 615
2.5.3 Probar el servidor . . . 617

3. El Display Manager . . . 617

3.1 Principio . . . 617
3.2 XDM . . . 618
3.2.1 Configuración general . . . 618
3.2.2 Setup: Xsetup . . . 620
3.2.3 Chooser: RunChooser . . . 621
3.2.4 Startup: Xstartup . . . 622
3.2.5 Sesión: Xsession . . . 622
3.2.6 Reset: Xreset . . . 623
3.2.7 Resources: Xresources . . . 623
3.2.8 Servers: Xservers . . . 624
3.2.9 AccessFile: Xaccess y XDMCP . . . 624
3.3 gdm y kdm . . . 624
3.4 xdm, gdm o kdm al boot . . . 628
3.4.1 inittab . . . 628
3.4.2 Servicio System V . . . 629
3.4.3 Destino systemd . . . 629

www.fullengineeringbook.net

3.4.4 Service upstart . . . 630
3.4.5 /etc/sysconfig . . . 630
3.4.6 Ubuntu y Debian . . . 631

4. Window Manager y entorno personal . . . 632

4.1 Mediante el Display Manager . . . 632
4.2 startx . . . 633
4.3 Los terminales . . . 634
4.4 Los gestores de ventanas . . . 635
4.4.1 twm . . . 635
4.4.2 IceWM . . . 635
4.4.3 fvwm . . . 636
4.4.4 CDE . . . 637
4.4.5 WindowMaker . . . 637
4.4.6 Enlightenment . . . 637
4.4.7 Xfce . . . 637
4.4.8 KDE y GNOME . . . 638
4.4.9 Los demás . . . 638
4.5 Exportar las ventanas . . . 639
4.6 Los recursos de una aplicación X . . . 640
4.6.1 Modificar la apariencia de un programa . . . 640
4.6.2 Editar los recursos . . . 642
4.6.3 xrdb . . . 646

5. Accesibilidad . . . 647

5.1 Asistencia al teclado y el ratón . . . 648
5.2 Asistencia visual y auditiva . . . 650

Particionamiento avanzado: RAID, LVM y BTRFS

1. Particionamiento avanzado RAID . . . 651

1.1 Definiciones . . . 651
1.2 Precauciones y consideraciones de uso . . . 652
1.2.1 Disco de emergencia . . . 652

www.fullengineeringbook.net

1.2.2 Disco averiado . . . 652
1.2.3 Boot . . . 652
1.2.4 Swap . . . 653
1.2.5 Periféricos . . . 653
1.2.6 IDE y SATA . . . 653
1.2.7 Hot Swap . . . 654
1.3 RAID con mdadm . . . 654
1.3.1 Preparación . . . 654
1.3.2 Creación . . . 655
1.3.3 Guardar la configuración . . . 657
1.4 Estado del RAID . . . 657
1.5 Simular una avería . . . 658
1.6 Sustituir un disco . . . 659
1.7 Apagado y puesta en marcha manual . . . 660

2. Introducción al LVM . . . 660

2.1 Principio . . . 660
2.2 Los volúmenes físicos . . . 662
2.2.1 Crear un volumen físico . . . 662
2.2.2 Ver los volúmenes físicos . . . 662
2.3 Los grupos de volúmenes . . . 663
2.3.1 Crear un grupo de volúmenes . . . 663
2.3.2 Propiedades de un grupo de volúmenes . . . 663
2.4 Los volúmenes lógicos . . . 665
2.4.1 Crear un volumen lógico . . . 665
2.4.2 Propiedades de un volumen lógico . . . 666
2.4.3 Acceso al volumen lógico . . . 667
2.5 Ampliaciones y reducciones . . . 668
2.5.1 Los grupos de volúmenes . . . 668
2.5.2 Ampliar un volumen lógico . . . 670
2.5.3 Reducir un volumen lógico . . . 673
2.5.4 Mover el contenido de un volumen físico . . . 674
2.5.5 Reducir un grupo de volúmenes . . . 676
2.6 Suprimir un grupo de volúmenes . . . 677
2.6.1 Etapas . . . 677
2.6.2 Suprimir un volumen lógico . . . 677

www.fullengineeringbook.net

2.6.3 Retirar todos los volúmenes físicos . . . 677
2.6.4 Destruir un grupo de volúmenes . . . 677
2.6.5 Suprimir un volumen físico . . . 678
2.7 Comandos adicionales . . . 678

3. Utilización extendida de BTRFS . . . 679

3.1 Los subvolúmenes . . . 679
3.1.1 Un sistema de archivos dentro de otro sistema de archivos . . . 679
3.1.2 Creación . . . 679
3.1.3 Montaje . . . 680
3.1.4 Destrucción . . . 681
3.2 Los snapshots . . . 681
3.2.1 Fundamentos . . . 681
3.2.2 Creación . . . 681
3.2.3 Montaje . . . 682
3.2.4 Destrucción . . . 682
3.2.5 Operaciones con los ID . . . 682
3.3 Utilizar varios discos . . . 683

Anexo

1. Modelo SQL (para el capítulo El shell y los comandos GNU) . . . 685

índice . . . 689

www.fullengineeringbook.net

LINUX

Dominar la administración del sistema (3ª edición)

Este libro sobre **la administración de sistemas Linux** se dirige a todo informático que vaya a gestionar este sistema operativo y que desee aprender o consolidar los conocimientos básicos adquiridos sobre la materia.

Sea cual sea la distribución de Linux empleada (bien sea en entornos empresariales o domésticos), se presentan y detallan todos los métodos y comandos de administración de Linux. Esta nueva edición del libro contempla las **últimas novedades de Linux**.

El libro realiza un recorrido por los conocimientos necesarios para **la instalación** de una distribución, la gestión de **paquetes** de software, la**compilación** a partir de las fuentes, las **librerías** compartidas, los principales comandos **Gnu** y los scripts shell, la gestión de **discos** y**sistemas de archivo**, la implementación de **volúmenes RAID y LVM**, el**arranque y parada** del sistema, **la impresión** y las tareas comunes de administración de la gestión de **usuarios** y **la automatización de tareas**, la configuración de **redes** y los servicios asociados, **el núcleo** y su compilación, las bases de la **seguridad** o la configuración **del entorno gráfico**.

Todos los puntos tratados se complementar con ejemplos y su dominio le convertirá en un administrador de sistemas Linux competente.

Los capítulos del libro:
Introducción – Presentación de Linux – Instalación de Linux y de los paquetes de software – El shell y los comandos GNU – Los discos y el sistema de archivos – Inicio de Linux, servicios, núcleo y periféricos – Las tareas administrativas – La red – La seguridad – X Window – Particionamiento avanzado: RAID, LVM y BTRFS – Anexo

---

Sébastien ROHAUT

Después de pasar 15 años en una empresa de ingeniería informática**Sébastien ROHAUT** es hoy en día Ingeniero de Sistemas en un importante grupo Francés. Asimismo, durante más de 10 años dio cursos de preparación de ingenieros, en concreto en el marco de la preparación para los exámenes de Linux. Esta gran experiencia técnica y pedagógica le permite ofrecer a los lectores un libro eficaz sobre la administración de un sistema Linux.

Introducción

En los últimos 25 años, Linux ha pasado de ser el hobby de estudiantes a una verdadera solución empresarial. Para un administrador de sistemas Unix es imposible no conocer este sistema operativo. En 2014, Linux representaba al menos el 40% de los servidores web, el 65% de los sistemas móviles y el 98% de los superordenadores.

Instalar Linux es muy sencillo y las tareas de administración comunes tienden también a serlo, porque la complejidad del sistema se oculta a menudo por los interfaces gráficos proporcionados por los fabricantes de las principales distribuciones. Sería un gran error especializarse en una u otra de estas distribuciones: sus interfaces no son más que frontales basadas en las mismas herramientas que modifican los mismos archivos de configuración, por lo que más bien hay que esforzarse en conocer estos. En lugar de utilizar una interfaz que quedará obsoleta en la próxima versión, rompa su dependencia y aprenda a dominar directamente los misterios de su sistema.

Unix es fundamentalmente un sistema operativo que se instala en servidores, pero también, mediante Linux, en entornos domésticos: puestos de trabajo, smartphones, tabletas, "cajas" Internet... Tanto si Vd. es un simple usuario como si es un ingeniero de sistemas, puede comprender mejor el comportamiento de su sistema operativo si conoce su funcionamiento y su administración. Conocer el sistema es también conocer el funcionamiento intrínseco de un ordenador.

Este libro le permite aprender a administrar un equipo que funcione con Linux. Se abordan todos los puntos esenciales, desde la instalación básica hasta la administración avanzada, tanto de sistemas como de redes. Es una guía práctica, pero también un libro de consulta al que podrá dirigirse en caso de necesidad: manténgalo cerca de su escritorio en su trabajo o en casa, no será raro que lo consulte. Su contenido no es específico para una distribución: es válido para todas, y muestra tanto los puntos comunes como las diferencias entre ellas y particularmente entre las dos principales corrientes dominantes: Red Hat y Debian, al igual que sus correspondientes derivados.

Al terminar la lectura de este libro y poner en práctica los conocimientos adquiridos, estará preparado para gestionar la explotación y administración de puestos de trabajo y servidores bajo el sistema operativo Linux. Será, realmente, un administrador de sistemas Linux, una competencia muy apreciada en el mundo empresarial. ¡Feliz lectura!

Bienvenidos al mundo Unix

1. Un nuevo mundo

Linux no es una simple moda pasajera ni una campaña promocional. Desde sus primeros desarrollos en 1991 y hasta la fecha de hoy, Linux no ha dejado de evolucionar, cambiar. El mundo de la informática está vivo. Si no evoluciona, vegeta. Gracias a Linux, millones de personas han encontrado por fin lo que estaban buscando.

Linux no es más complicado que cualquier otro sistema operativo. La lentitud con la que parece penetrar Linux en el gran público no está relacionada con su supuesta dificultad. El buen rendimiento admitido por numerosos usuarios principiantes o avanzados, grupos de usuarios Linux y profesionales en general muestra que se trata de un problema relacionado con las costumbres de la gente, habituada durante años a un sistema operativo único. Resulta evidente que hay que modificar un poco estas costumbres para adaptarse a un entorno Linux, de la misma manera que conducir un coche familiar no le convierte en un as de la conducción deportiva con Ferrari.

2. Historia de los ordenadores

a. Complejidad de los ordenadores

Un ordenador es una máquina electrónica extremadamente compleja. Si bien el principio mismo del ordenador tal como lo conocemos no ha cambiado desde la época de Alan Turing o de Conrad Suze y data de principios de los años 40 e incluso antes (máquina de Charles Babbage), las evoluciones tecnológicas y la miniaturización han permitido crear máquinas cada vez más potentes y, al mismo tiempo, pequeñas. Desde los primeros ordenadores electromecánicos compuestos de miles de tubos de vacío y cuya programación se llevaba a cabo conectando cables, al ordenador moderno de hoy, la complejidad del hardware ha sido creciente. Hubo una época en la que un ordenador ocupaba toda una planta y el circuito de agua que permitía enfriarlo (algunos eran enfriados mediante bloques de hielo) calentaba todo un edificio. Hoy en día basta con irse de compras al supermercado del barrio para adquirir un ordenador: se podría pensar de manera lógica que la sencillez en la adquisición de uno de estos equipos traería aparejada la sencillez de uso, y que el ordenador debería ser tan fácil de utilizar como su lector de DVD de salón.

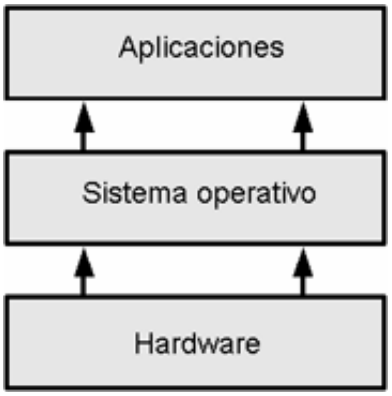
Desafortunadamente, no es el caso. Un ordenador es una máquina complicada. ¿Debemos esperar que sea tan sencillo de usar como una máquina de mecanismo simple mientras que un ordenador es un equipo completamente polivalente? ¿Debemos conocer a toda costa los entresijos técnicos de nuestro ordenador para poder utilizarlo? La respuesta varía según el uso y las personas, pero suele ser negativa. Sin embargo, hay que recordar y respetar algunas nociones y bases fundamentales.

b. La inteligencia

Un ordenador no sabe hacer nada por sí mismo. No hay nada más estúpido que una computadora. Hay que decirle lo que debe hacer. El ordenador es menos inteligente que cualquier insecto. Lo que le hace "inteligente" en el sentido de potencia de cálculo es usted y los programas que le hace ejecutar. Aunque esté inventado por humanos para humanos, el ordenador no entiende su lenguaje. Habla en binario, combinación de ceros (0) y unos (1). Estos valores combinados entre sí forman palabras y datos para el ordenador. El binario se convierte en un lenguaje llamado "lenguaje de máquina". Los microprocesadores emplean un lenguaje llamado "ensamblador" en el cual cada instrucción dispone de un equivalente en binario. Con la ayuda de este lenguaje ensamblador, los informáticos desarrollan sus programas, uno de los cuales se llama "compilador": un traductor de lenguaje de alto nivel y comprensible esta vez por un gran número de informáticos.

3. El sistema operativo

Entre el momento en el que pulsa el botón de encendido de su ordenador y el momento en el que puede por fin trabajar y utilizar sus programas, pasa un cierto tiempo durante el cual se cargan programas en la memoria del ordenador. El objetivo de estos programas es simplificarle la vida al usuario y al programador, al hacer las cosas más sencillas. Estos programas forman un conjunto llamado "sistema operativo". Como indica su nombre, el papel del sistema operativo consiste en sacar partido al ordenador tanto como sea posible. En otras palabras, el sistema operativo le proporciona todos los mecanismos necesarios para aprovechar lo mejor posible los recursos de su ordenador.



Fundamentos del sistema operativo

Un sistema operativo es un programa o un conjunto de programas que garantiza la gestión del ordenador y de los periféricos. Sirve de interfaz entre el medio físico (hardware) y el programa (software). Es un conjunto de programas muy complejos cuya meta es hacer más sencillos los programas y el uso del ordenador.

El sistema operativo ofrece a los programadores una interfaz de programación de aplicaciones llamada **API**, *Application Programming Interface*. Todos los programadores utilizan las mismas funciones en sus programas, lo que simplifica mucho el trabajo. Pueden centrarse en el objetivo de su programa (crear un procesador de texto, por ejemplo) sin tener que escribir una y otra vez las secuencias de programas que gestionan el disco duro, la impresora o el acceso al teclado.

Le corresponde al sistema operativo gestionar:

- la memoria,
- los accesos a los periféricos,
- los datos en los discos,
- los programas,
- la seguridad,
- la recopilación de la información.

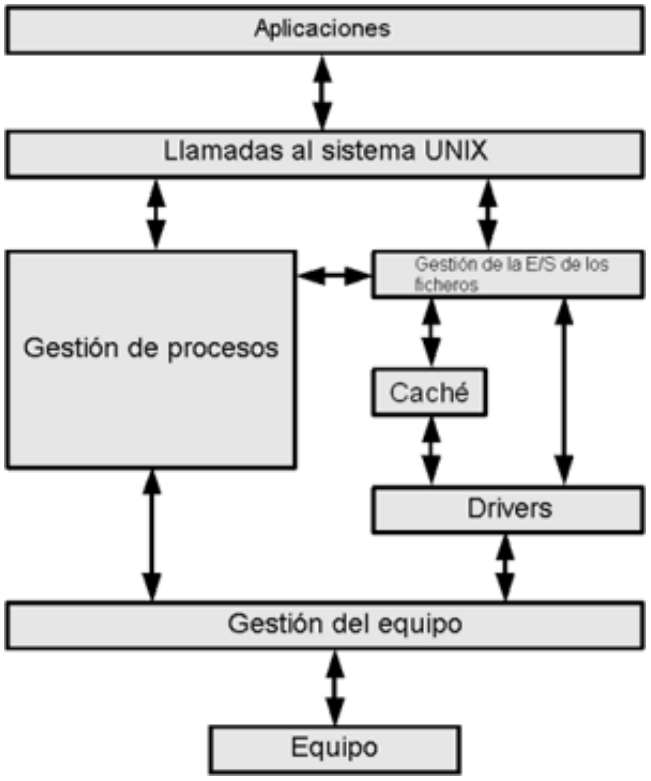
Falta aún la interfaz gráfica. En un producto como Microsoft Windows la interfaz gráfica está incluida en el sistema operativo. Desde hace mucho tiempo es imposible trabajar sin ella, cualquier ajuste en el sistema se realiza desde un cuadro de diálogo. Los usuarios suelen pensar que la interfaz gráfica es el sistema operativo. Sin embargo, la interfaz gráfica no siempre ha formado parte del sistema operativo. Viene como complemento. No encontrará ningún libro sobre teoría de los sistemas operativos que trate de las interfaces gráficas. ¿Por qué? ¿Cuál es el interés, más allá de aumentar la lentitud y ocupar preciosos recursos de la máquina, de tener una interfaz gráfica para gestionar un servidor de Internet? Linux ofrece interfaces, pero son programas como los demás.

La interfaz gráfica no es un componente del sistema operativo y Linux no la necesita para

funcionar correctamente. Es un conjunto de programas ejecutados de forma clásica "sobre" el sistema operativo.

Linux es un sistema operativo de tipo Unix. Existen decenas de sistemas operativos en esta familia. Unix es un sistema operativo de la familia de los sistemas **multitarea** y **multiusuario**:

- **Multitarea**: el sistema gestiona la ejecución simultánea de varios programas llamados procesos (nota: un verdadero sistema multitarea necesita contar con varios microprocesadores o equivalentes; Hyper Threading, por ejemplo).
- **Multiusuario**: el sistema permite la existencia de varios usuarios diferentes en una misma máquina, conectados o no (un usuario puede ejecutar un programa sin estar conectado, como por ejemplo un servidor de Internet).



Arquitectura lógica de un sistema UNIX

El esquema anterior es una síntesis simplificada de la estructura interna de un sistema operativo Unix. Abajo se encuentra el hardware; encima, los programas que ejecuta su equipo. Entre ambos, los diferentes componentes del sistema operativo garantizan su correcto funcionamiento:

- Las **llamadas al sistema** son utilizadas por los programas para comunicarse con el sistema operativo Unix.
- La **gestión de los procesos** se encarga de la conmutación de las tareas y de su prioridad. Este componente se encarga, por lo tanto, de la multitarea.
- La **gestión de las entradas y de las salidas de archivos** se encarga tanto de la lectura y escritura de los datos en sus discos duros como también en sus periféricos (tarjeta de sonido, impresora, etc.).
- Se puede ubicar cierta información en una zona de memoria llamada **caché**. En lugar de escribir datos directamente en el disco duro (un proceso lento), Unix va a escribirlos provisionalmente en una zona de memoria para luego escribirlos en el disco, después de unos segundos. Así, la lectura de estos datos es más rápida porque ya están en memoria y el programa no pierde tiempo en esperar el fin de la escritura de los datos.
- Los **controladores** tienen como papel gestionar al más bajo nivel el hardware o las estructuras lógicas del hardware (por ejemplo, los datos de una partición).

Una aplicación bien programada en un sistema operativo bien programado no puede cortocircuitar este esquema: no "discute" nunca con el hardware, sino que dialoga obligatoriamente con las API correspondientes.

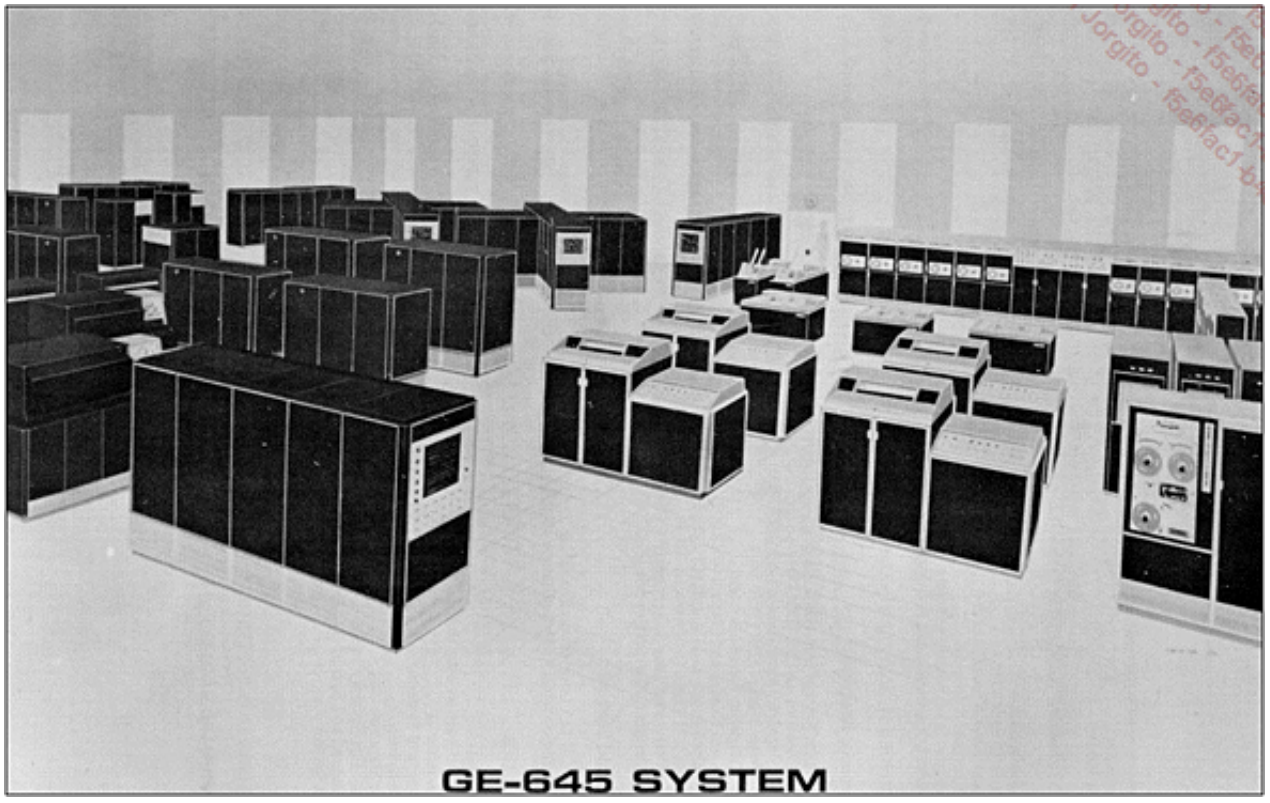
4. El sistema Unix, una breve historia

a. De MULTICS a UNIX

La historia de Unix empieza en **1964**, cuando el MIT, el laboratorio Bell Labs de AT&T y General Electric comienzan a desarrollar el proyecto experimental **MULTICS** (*Multiplexed Information and Computing Service*). El proyecto Multics responde a estas nuevas necesidades:

- posibilidad de ser utilizado por varias personas a la vez,
- posibilidad de lanzar procesos en segundo plano,
- una gestión más adecuada de la seguridad.

Multics se desarrolla en un enorme sistema GE-645 de General Electric, equipado con dos procesadores que pueden tratar cada uno 435 000 instrucciones por segundo, tres unidades de memoria de 1 MB cada una y 136 MB de almacenamiento. Funcionó en el MIT hasta 1988, con 82 estaciones de trabajo, y un máximo de 200 usuarios y llegó a trabajar de forma simultánea en General Electric. La última instalación Multics que ha sido desactivada es la del Ministerio de Defensa Canadiense, el 30 de octubre de 2000.



El GE-645

Sin embargo, si bien Multics alcanzó pronto un grado de estabilidad lo suficientemente aceptable como para pasar a producción, resultó que su rendimiento era menor de lo esperado. En **1969** Bell Labs abandona el proyecto para dedicarse al desarrollo de otro sistema llamado GECOS.

Ken Thompson, desarrollador en Bell, continuó trabajando en el GE-645 y se propuso crear un juego llamado Space Travel. Al ejecutarse en Multics resultó ser muy lento y costoso, más si se le

media en tiempo compartido (antes de la multitarea, el tiempo de máquina era seccionado y cada una de estas era contabilizada y facturada).

Ken Thompson

Ken reescribe entonces el juego en un ensamblador para el miniordenador DEC PDP-7. En esta misión le ayuda **Dennis Ritchie**, procedente también de Bell Labs. Esta experiencia, combinada con la obtenida durante el diseño de Multics, empuja a los dos hombres y a su equipo a crear un nuevo sistema operativo para el PDP-7. **Rudd Canaday**, también de Bell Labs, estaba desarrollando justamente un nuevo sistema de archivos que diseña como si fuera un sistema operativo; de ahí que Unix sea un sistema orientado a archivos, en el cual todo (o casi) es un archivo. Le añadieron un intérprete de comandos y algunas herramientas. Llamaron al sistema **UNICS** (*Uniplexed Information and Computing System*), según una idea de **Brian Kernighan**. Desde el principio el proyecto podía gestionar dos usuarios al mismo tiempo en modo multitarea real.

La DEC PDP-7

El origen de la palabra es tema de numerosas leyendas, todas las cuales tienen probablemente su parte de verdad. UNICS es un juego de palabras en clave de humor de MULTICS, cuya arquitectura es el motivo de numerosas críticas en esta época:

"MULTICS (múltiple) hacía la misma cosa de varias maneras, mientras UNICS (único) hacía cada cosa de la misma manera". Además, en inglés UNICS se pronuncia como "eunuchs", o sea, "eunuco", un sistema Multics "castrado".

UNICS retoma los conceptos esenciales desarrollados para MULTICS y los mejora. En particular, sus diseñadores proponen un sistema de comunicación completamente nuevo entre los programas, en el cual un primer programa puede enviar sus datos a otro programa. Rápidamente, el *CS* fue sustituido por un *X*, una letra menos para la misma presentación. La leyenda **UNIX**acababa de nacer.

Todo hubiera podido finalizar aquí, ya que los equipos trabajaban sin financiación. Bell Labs se deshizo de Multics y de sus sucesores. Para proseguir los trabajos, Thompson y Ritchie proponen a Bell Labs añadir un programa de procesador de textos para el PDP-11/20 en UNIX. La compañía acepta, se pone a disposición la máquina y el equipo obtiene finaciación y respaldo oficial. La herramienta *runoff* (que se convertirá en roff y luego en troff) y el editor *ed* son desarrollados y por primera vez en **1970** se emplea la denominación **Unix Operating System**. Bell utiliza entonces Unix como un sistema de procesador de texto para la redacción de sus patentes. El primer manual de programación Unix data del 3 de noviembre de 1971.

b. El lenguaje C

Rápidamente, los ingenieros deben enfrentarse a un nuevo problema. Desarrollado en ensamblador y por lo tanto en lenguaje máquina, Unix debe ser reescrito en parte para cada nuevo modelo de ordenador DEC, y programar en ensamblador es un arte difícil. Es el momento de enfrentarse al reto de la portabilidad. En 1970, Thompson se interesa por el problema. Primero piensa en desarrollar Unix en lenguaje TMG y luego en Fortran. Al encontrar el lenguaje incompleto, se asocia con Dennis Ritchie para crear el **lenguaje B**, que proviene del lenguaje **BCPL**. Pero tampoco les parece el más adecuado (problema con el tipado de las variables y los números reales). Ritchie parte del lenguaje B y desarrolla el **New B**, que llama lógicamente el **lenguaje C**. C es volcado a lenguaje máquina tras pasar una etapa de compilación. La escritura de los programas resulta más rápida.

Dennis Ritchie

Unix es reescrito en lenguaje C a partir de **1973**. Para pasar un sistema Unix de una máquina a otra, basta con que haya un compilador C instalado en la nueva máquina. Es mucho más sencillo y eficiente escribir un compilador C (él mismo escrito en gran parte en C) que escribir de nuevo todo un sistema operativo en ensamblador. Únicamente los elementos que tenían mayor contacto con la arquitectura física de la máquina fueron escritos directamente en lenguaje máquina. Unix se hace portable y eso permite acelerar su desarrollo.

c. Las licencias y el advenimiento de BSD y System V

Un hecho de gran calado va a contribuir en aquel entonces a consolidar la gran difusión de Unix (la

palabra "gran" puede parecer exagerada cuando se está hablando de decenas de copias del sistema). AT&T, de la cual depende Bell Labs, fue objeto en 1956 de un decreto antimonopolio que le prohibía comercializar cualquier producto ajeno a su sector industrial: las telecomunicaciones. No puede vender Unix. AT&T (que ni siquiera le ve futuro comercial) decide en **1974** liberar completamente el sistema UNIX, pero sólo con fines educativos a las universidades y las empresas y bajo una licencia que resulta ser muy poco restrictiva. Sólo el código fuente (*el texto del programa no compilado aún*) del núcleo en ensamblador no se incluirá en la liberación, al menos oficialmente. UNIX cuenta ya con algunas versiones, de entre las cuales las más difundidas son la sexta, de**1975**, y la séptima, de **1978**. La siguiente versión, Unix v7, fue la primera específicamente diseñada con la intención manifiesta de ser llevada a otras máquinas además de las PDP, en particular al modelo VAX 11/780. Se considera que la v7 fue la última versión completamente común a todos los Unix siguientes.

Por aquel entonces se produce otro gran acontecimiento. Unix iba a celebrar sus diez años y las universidades estadounidenses se estaban implicando con fuerza en su difusión y mejora cuando AT&T modifica la licencia de Unix haciéndola más restrictiva. La entidad que comercializa Unix tiene la autorización de vender licencias del código fuente. Las tarifas, prohibitivas, obligan a las universidades a continuar, para lo bueno y lo malo, sus desarrollos tomando como punto de partida las versiones anteriores al cambio de la licencia. Una de estas universidades se encuentra en California y se llama **Berkeley**. Berkeley es la mayor contribuidora de Unix desde que empezó a trabajar en él, en 1974. La primera versión UNIX de **BSD** (*Berkeley Software Distribution*) está basada en Unix v6, de **1977**, y recibe el nombre comercial de **1BSD**. La versión **2BSD** está basada en Unix v7 y data de 1978.

A partir de ahí, se asiste al enfrentamiento de dos escuelas UNIX. La primera, en teoría la oficial, es la de AT&T, que va a seguir desarrollando las versiones 8, 9 y 10 durante los años 80 del siglo XX, con fines científicos. De forma paralela, desarrolla un Unix totalmente comercial llamado **Unix System III**, que se comercializa a partir de **1982**. En **1983** AT&T desarrolla y vende las primeras versiones Unix System V. La última versión, **Unix System V release 4.2**, data de **1993**. Esta versión de Unix se conoce con la abreviatura SVR4 y su principal característica es que su código fuente está disponible bajo licencia. Esto significa que un organismo está autorizado a comprar una y a desarrollar su propia versión comercial.

Durante este tiempo, la universidad de Berkeley no descansa y sigue desarrollando BSD como alternativa Open Source al Unix System III y V, cuyas fuentes no puede usar por carecer de derechos. Es en BSD donde se va a implementar por primera vez el protocolo **TCP/IP**, base del Internet moderno, gracias a la financiación del Ministerio estadounidense de Defensa. La última versión oficial de BSD es la **4.4BSD** y data de junio de **1994**.

d. La guerra de los Unix

El período que va desde la mitad de los años 80 hasta 1994 no ha estado exento de sobresaltos. Los efectos de la separación de Unix en dos ramas ha resultado ser un desastre y a punto ha estado de significar su pérdida total. Los dos campos (AT&T con su System V y Berkeley con su BSD) no se ponen de acuerdo en un estándar común. La consecuencia, además de la celebración de múltiples juicios (hasta 1993) sobre el uso del nombre y herramientas derivados de Unix, es el hecho de que han crecido como la mala hierba multitud de versiones de Unix comerciales y sobre todo incompatibles entre ellas. Es a partir de esta época cuando surgen los grandes nombres de los clones Unix, entre los cuales cabe destacar **Solaris**, **AIX**, **OSF1** / **Digital Unix** / **True64**, **Xenix**,**HP-UX**, **IRIX**, **Ultrix**, **Unixware**, **A/UX**. Todos son incompatibles con el vecino, pero claman alto y claro su pertenencia a Unix. Esta guerra de los Unix se conoce realmente como el período oscuro de las *Unix wars*. Pero lo más triste es que nadie parece interesado en ponerse de acuerdo sobre una base y un estándar común. El efecto directo de esta guerra ha sido la creación de un vacío en el mercado aprovechado por Microsoft para colocar con fuerza su sistema operativo **Windows NT**(que a su vez, pero poca gente lo sabe, deriva de Unix).

En **1984** un grupo de editores comerciales de Unix intenta una primera estandarización creando X/Open Standards y publicando un documento llamado **X/Open Portability Guide**, que describe un estándar abierto (accesible a todos) para Unix. Este comité obtiene un gran espaldarazo en **1987**, cuando Sun Microsystems y AT&T deciden trabajar sobre un Unix unificado, fusión de BSD y de System V. El resultado es en realidad System V Release 4.

El refrán dice que pican más los celos que las pulgas. La competencia acusa a Sun de dirigir el cotarro y funda *Open Software Foundation*, o sea, **OSF**, en **1988**. OSF declara que también persigue un estándar abierto para Unix, pero basado de manera exclusiva en BSD. Publicando sus especificaciones en **1990**.

Como respuesta, AT&T y su grupo crean *Unix International* en **1989**, en un enésimo intento de unificación, sin éxito. En medio de este embrollo, AT&T decide deshacerse de Unix, del que sigue siendo propietario oficialmente, y crea para ello una empresa llamada *Unix System Laboratories* en**1992**. Todos los derechos de Unix se transfieren a USL.

e. La estandarización

Un año más tarde, y cuando la situación amenaza con estancarse, aparece un nuevo actor que pretende triunfar donde los otros fracasaron. La empresa **Novell** compra USL el año de su creación y se convierte en el propietario de SVR4.2. En **1993** Novell cede la marca Unix a X/Open. Unix International desaparece en **1994** y se reestructura OSF. Finalmente, en **1995** Novell cede la licencia operativa del código fuente de Unix a la empresa **SCO** *Santa Cruz Operations* (que se convertirá en Caldera y luego, de nuevo, en SCO). El mismo año X/Open y OSF se fusionan de manera definitiva y forman **The Open Group**.

Ya sólo existe un único organismo de estándar Unix. Unix es, por lo tanto, un sistema operativo abierto: sus especificaciones son conocidas y cada editor de sistema Unix comercial o gratuito que desea asegurar una compatibilidad con el conjunto de los Unix debe implementar este estándar, aunque sigue siendo libre de programar este estándar como desee, ya que una misma función puede escribirse de varias maneras. Los "estragos" de las *Unix Wars* han sido numerosos y han permitido el surgimiento de otros sistemas operativos como Windows NT de Microsoft.

f. Unix es un estándar

Para asegurarse de que todos los Unix siguen las mismas recomendaciones, The Open Group publica las normas (*Single Unix Specification*, *Unix95*, *Unix98*, *Linux Standard Base*, etc.) y puede proponer certificaciones. Estas normas se apoyan, en parte, sobre las definidas desde **1988** por el**IEEE** *Institute of Electrical and Electronics Engineers* (que se pronuncia "I3E") y en particular en la**IEEE 1003**, llamada también **POSIX** (*Portable Operating System Interface*). La X es al mismo tiempo una herencia y un reconocimiento al trabajo efectuado en Unix. IEEE 1003 se compone de 15 documentos que agrupan por temas todo lo que debe contener un Unix (comandos básicos, intéprete de comandos, interfaces de usuario, funciones de programación, etc.) para estar conforme al estándar POSIX. POSIX no se limita a Unix. Windows NT es compatible con POSIX para algunos de sus componentes. POSIX no es un estándar abierto. Las especificaciones de *The Open Group* están abiertas y accesibles a todos y los editores prefieren referirse a ellas.

Para poder utilizarse en algunas administraciones estadounidenses, un sistema operativo Unix debe ser compatible con estándar POSIX. De hecho, cuando Linux tuvo que usarse, el gobierno de Bill Clinton hizo que el Tesoro estadounidense financiara la totalidad de la certificación PCTS (*Posix Conformance Test Suite*).

Las últimas versiones oficiales de BSD y System V datan de 1994. Los Unix concebidos a partir de 1995 implementan las recomendaciones de The Open Group. Sin embargo, históricamente algunas versiones siguen más "orientadas" a BSD, o más a System V en lo que se refiere a su configuración, o a veces a los dos, como Linux (y según la distribución).

g. Unix en los ordenadores personales

El primer Unix para ordenador personal, en el sentido de ordenador de tipo IBM PC es **Xenix**. Proviene de Unix v7 y salió en **1983** para PC (algunas versiones han estado disponibles incluso antes en otras arquitecturas físicas). Microsoft ha llevado Xenix, a costa de numerosas modificaciones, a otras máquinas. La versión 2 de Xenix data de 1985 y está basada en Unix System V. Cuando en 1987 IBM lanzó el sistema operativo OS/2 en asociación con Microsoft, este

último transfirió los derechos de Xenix a SCO. La versión 2.3.1 de este mismo año soporta el 386, SCSI y TCP/IP. Xenix se convierte en SCO Unix en 1989 y luego desaparece en favor de SVR4.

El norteamericano **Andrew Stuart Tanenbaum** (apodado Andy) es investigador y profesor de informática de la Universidad libre de Ámsterdam. También es autor de obras de referencia en informática sobre la teoría de los sistemas operativos. En **1987**, con fines pedagógicos, concibe y escribe el sistema operativo Minix. Utiliza 20 MB de espacio en disco y necesita pocos recursos (2 MB de memoria física). Minix tendrá una gran importancia para Linux. Minix sigue existiendo y la versión 3 salió en octubre de 2005.

Andrew Tanenbaum

Hay otras versiones de BSD adaptadas al PC. Una de ellas es el **386BSD**, de octubre de 1989 y que deriva de 4BSD. Su sucesor más famoso es el **NetBSD**, que deriva de 4.3BSD y de 4.4BSD. NetBSD es el sistema operativo más portado a otras arquitecturas físicas. **OpenBSD** está basado en 4.4BSD y muy orientado a la seguridad: en ocho años sólo se pudo encontrar un fallo de seguridad. **FreeBSD** deriva también de 4.4BSD y proviene directamente de la época de los juicios entre BSDI y AT&T. Gracias a ello, FreeBSD es completamente libre y abierto.

Solaris, el Unix de Sun Microsystems, está disponible desde hace varios años en PC y la versión OpenSolaris (versión 10) es Open Source.

Por último, **Linux** es sin duda el Unix libre más conocido y más extendido en el PC. La historia de su creación merece destacar algunos detalles adicionales.

El software libre

1. Los orígenes del software libre

Unix es el perfecto ejemplo del trabajo que se puede efectuar cuando se encauzan todas las energías a la búsqueda de un ideal tecnológico. Cuando AT&T distribuye casi libremente en 1974 el código fuente del sistema operativo a las universidades porque, entre otras razones, no ve ningún futuro económico a su producto, no parece dudar del entusiasmo de los estudiantes, profesores e investigadores en informática. Esta primera comunidad pasará mucho tiempo modificando y mejorando el producto, subiendo todas las novedades a AT&T para que se integren al producto oficial. Tras el cambio de licencia en 1978, la energía de la comunidad se encauzó hacia el proyecto universitario BSD, dejando el Unix comercial de AT&T. Señalemos que los más grandes progresos se hicieron con el Unix de Berkeley.

Los primeros ordenadores eran esencialmente herramientas de investigación en manos de universitarios (y también monstruos de cálculo para necesidades militares). En los laboratorios de investigación, los programas circulaban como las ideas: libremente. Era absolutamente normal que un programa desarrollado por un equipo de programadores o investigadores se distribuyera a otros equipos de otras universidades y a cualquier otro lugar donde hiciera falta.

Y nada había de raro en que este programa fuera modificado por otro equipo, y así sucesivamente. A día de hoy, cuando un ilustre matemático demuestra un teorema difícil, publica el resultado de sus investigaciones en obras especializadas con el fin de ayudar al progreso de la ciencia. Todo el mundo tiene acceso a ello.

Pero el universo de la informática ha seguido otros derroteros. Pese a ser una ciencia, el fruto de las investigaciones en informática no se circunscribe al mundo de los universitarios. Rápidamente, las empresas vieron el inmenso interés de automatizar algunas de sus tareas, como la contabilidad, los pagos, etc. Con la compra de los primeros grandes ordenadores de gestión, se necesitaron programas. Estos programas tuvieron que ser protegidos como secretos industriales, había nacido una nueva industria: el desarrollo de programas. Con su entrada en la dinámica de las grandes empresas, la informática perdió rápidamente la inocencia y se hizo mucho menos libre. Se empezó a hablar de licencias, impuestos y tasas, derechos de autor (lo que no impide autorizar la copia según el caso), limitación de los derechos, prohibición de copiar, etc.

2. El proyecto GNU y la FSF

Richard Stallman no fue seguramente el primero en lamentar este hecho, pero decidió reaccionar. Informático en el laboratorio de inteligencia artificial en el MIT a finales de los años 1970, emplea una impresora que se avería con frecuencia. Como él y sus compañeros disponían del código fuente del controlador (programa de gestión) de la impresora, lo modificaron para recibir una señal con cada avería. En un momento dado, el laboratorio compra un nuevo modelo de Xerox más fiable, pero el controlador para su sistema operativo no es proporcionado. Como desea adaptar este controlador a sus necesidades, Richard Stallman, recurre a otro laboratorio que dispone del código fuente, pero que se niega a proporcionárselo: Xerox lo prohíbe. Esto significa que la impresora nunca funcionará, y Stallman, muy contrariado por esta situación, decide obrar en pro de la defensa y la difusión del software libre enfrentándose al cerrado mundo del software propietario.

Richard Stallman

Stallman decide en **1983** escribir un nuevo sistema operativo de acceso, uso, modificación y redistribución completamente libres. Basado en Unix, lo nombra **GNU** (*Gnu's Not Unix*). Los acrónimos recurrentes estaban muy de moda entre los informáticos. Puede encontrarse el anuncio del proyecto y de las motivaciones de Stallman en http://www.gnu.org/gnu/initial-announcement.html.

Para su sistema necesita un núcleo (el corazón del sistema operativo) y herramientas (para gestionar los archivos, por ejemplo). Pero no se trata de su primera intentona, ya que escribió un gran editor de texto llamado **Emacs**. Los primeros desarrollos van muy deprisa y las herramientas son muy numerosas y a menudo de mejor calidad que las comerciales. A cambio, el diseño de un núcleo Unix es mucho más complejo y necesita una fase teórica importante. Se lanza el proyecto**HURD** (*Hird of Unix Replacing Daemons*), que aún no ha finalizado.

La batalla no es solamante técnica, sino también política, filosófica, comercial y jurídica. Para defender el software libre, Stallman crea la **FSF** (*Free Software Foundation*) en 1985, que publica las ideas del software libre. Entre sus primeros trabajos figura la redacción (con la ayuda de abogados) de una licencia especial para estos programas llamada la **GPL** (*General Public License*). Un software libre garantiza cuatro libertades:

- **Libertad 0**: la libertad de utilizar un programa, sea cual sea el uso que se haga de él.
- **Libertad 1**: la libertad de estudiar el funcionamiento del programa y adaptarlo a sus necesidades.
- **Libertad 2**: la libertad de redistribuir copias con el fin de ayudar a su vecino (en el sentido amplio del término).
- **Libertad 3**: la libertad de mejorar el programa y difundir las mejoras al público general con el fin de que pueda aprovecharlo la comunidad.

Las libertades 1 y 3 necesitan tener obligatoriamente acceso al código fuente del programa. La libertad 3 define la noción de comunidad alrededor del software libre.

Observe que la palabra "gratuito" no se indica en ninguna parte. En inglés "free" significa tanto libre como gratuito. Se debe entender el software libre en el sentido de "libertad", y no de "gratuidad" (Free as a speech y no Free as a beer, como dicen en inglés). Es totalmente posible, y a veces aconsejable, empezar con el software libre. Pero como las libertades 2 y 3 autorizan la publicación del programa, siempre es posible recuperar una copia gratuitamente, y de manera legal. La gratuidad es un efecto de la libertad tal como está definida para el software libre.

Los trabajos de HURD avanzan poco o mal. Sus desarrolladores han apostado por desarrollar un micronúcleo, mientras que los subsistemas se hallan "esparcidos" en varias unidades independientes, pero que deben comunicarse entre ellas. La elección teórica es excelente, pero la implementación técnica es muy difícil. GNU no dispone de núcleo. Linux finalizará el proyecto en**1992**, cuando su licencia se convierte al GPL.

3. El open source

¿Y qué pasa con el **open source**? La expresión apareció en 1998 cuando Netscape Communicator se convirtió en un programa libre. La expresión *Open Source* (código abierto) era utilizada en las campañas publicitarias que intentaban llegar a las empresas con la publicación del código fuente libre. El reto consistía en hacer abstracción de las aportaciones fundamentales del código fuente libre para centrarse únicamente en las ventajas técnicas y económicas de este nuevo modelo. Con el tiempo, la expresión ha sido retomada en todos los sentidos por los medios de divulgación y las empresas, y su definición se ha "emborronado". Se habló de "open source limitado", proponiendo el acceso a las fuentes pero sin derecho de modificación o de redistribución. En nuestra opinión, el programa libre no debe mostrar medias tintas. O es libre o no lo es.

Si quiere estar seguro de que el programa que utiliza es libre, compruebe el nombre de la licencia y diríjase al sitio de **OSI** "*Open Source Initiative*", http://www.opensource.org, donde se recogen la mayoría de los más conocidos. Es una iniciativa de **Eric S. Raymond** (ESR), hacker (especialista de alto nivel) y uno de los grandes nombres del open source. A veces en conflicto con Richard Stallman, sus dos visiones (técnicas para ESR, filosóficas para Stallman) son, sin embargo, complementarias.

4. GNU/Linux

a. Linus Torvalds

La historia de Linux empieza cuando Linus Torvalds, joven estudiante finés de la universidad de Helsinki de 21 años de edad, adquiere en 1991 un ordenador 386 para sustituir su Sinclair QL que empieza a mostrar limitaciones. El 386 es un microprocesador de 32 bits genial que gestiona, entre otras cosas, la memoria virtual y la conmutación de las tareas. Pero el gran problema es que recibe un PC con MS-DOS, un sistema operativo que está lejos de ser ideal y sobre todo que no saca el rendimiento adecuado a este procesador. Linus tuvo entonces la idea de instalar otro sistema llamado Minix, un pequeño Unix sencillo y gratis desarrollado por el famoso Andrew Tanenbaum, que sí aprovechaba su bonito PC recién comprado a crédito. Linus se pone a trabajar y a desarrollarlo. Su objetivo es aprender el funcionamiento del 386, en particular la conmutación de las tareas en lenguaje ensamblador. Empieza a trabajar en un proyecto bastante sencillo: un emulador de terminal, completamente en ensamblador, para conectarse al servidor de su universidad.

Linus Torvalds

b. El accidente

Pero he aquí que un día borra accidentalmente los primeros sectores de la partición de su disco duro que contiene Minix, borrando por lo tanto su principal herramienta de desarrollo. Sólo le quedan dos soluciones: o reinstalarlo todo, o partir de lo aún queda y tratar de mejorarlo y aumentar su autonomía. Por supuesto, el entorno de desarrollo es reinstalado, pero Linus decide mejorar su proyecto y añadirle lo necesario: código básico, controlador rudimentario de disco duro, paso al lenguaje C, etc. El 25 de agosto de 1991, la versión 0.01 está lista y se publica ante una casi total indiferencia. Para las herramientas, nada más sencillo: el proyecto GNU iniciado por Richard Stallman dispone ya de todo lo necesario. Linux será el núcleo que falta al sistema operativo GNU.

c. La primera versión oficial

El objetivo de Linux es superar a Minix. Ya en su primera versión, es preciso que cuente con un shell (intérprete de comandos) y el gcc (compilador C). Esto lo consigue con la versión 0.02, anunciada en estos términos el 5 de octubre de 1991 en el grupo com.os.minix:

"¿Echa de menos los días felices de Minix-1.1, época bendita, cuando los hombres eran dignos de este nombre y escribían sus propios controladores de periféricos? ¿Quiere comprometerse en un proyecto original y está deseando encontrar un sistema modificable a su conveniencia? ¿Está deseando que todo funcione bajo Minix? ¿Echa de menos las noches en blanco intentando implementar un programa recalcitrante? Si es así, lea lo que sigue: como ya he publicado hace un mes, estoy trabajando actualmente en una versión libre de un sistema similar a Minix para un ordenador AT-386. Este sistema ya está operativo (aunque tal vez no le convenga, todo depende de lo que esté buscando), y pienso publicar sus fuentes. Se trata, de momento, de la versión 0.02, capaz sin embargo de ejecutar bash, gcc, gnu-make, gnu-sed, compress, etc."

d. El éxito comunitario

A partir de este momento y gracias a la publicación por Internet, Linux conoce el éxito y las contribuciones empiezan a llegar. Nace la comunidad Linux. La versión 0.03 ve la luz, y luego la 0.10. En 1992, Linux puede por fin hacer funcionar la interfaz gráfica X11. De hecho, se cree que es momento de pisar el acelerador y la versión siguiente recibe el nombre de 0.99 pensando que ya ha llegado a la meta. Fue un error. En efecto, hubo que esperar 18 meses antes de que la versión 0.99pl114 estuviese finalizada y de que por fin la versión 1.0 saliera, en enero de 1994. Entre los primeros balbuceos y la versión 1.0 hubo algunas sorpresas, en particular un conflicto "técnico" entre la concepción monolítica de Linux y la visión micro-núcleo de Andrew Tanenbaum. Este último comenzó las hostilidades con la famosa frase: "Linux está obsoleto".

e. Los años 1994-1997

Estos años ven aparecer las grandes distribuciones Linux que conocemos todavía hoy en día: Red Hat, Debian, SuSE, Slackware, Mandriva (anteriormente Mandrake). Durante estos años, Linux no deja de mejorar, con la llegada de la modularidad y de la versión 2.0. Es sobre todo durante estos años cuando Linux sale del pequeño mundo de los hackers y se da a conocer en las empresas. Los proyectos abundan, y la idea de mejorar el sistema y prepararlo para el mundo del escritorio (desktop) se abre camino con el principio del desarrollo de productos como Gnome o KDE.

La mascota de Linux, llamada **Tux**, data de 1996 y fue creada por Larry Ewing con la ayuda del programa libre GIMP. Tux (apócope de *Tuxedo* y *Torvalds UniX*) no es un pingüino, sino un *pájaro bobo pigmeo*. El hecho es que la palabra inglesa *penguin* designa en este idioma tanto al verdadero pingüino (*razorbill*) como al pájaro bobo; de ahí la confusión.

Tux, la mascota de Linux

f. A partir de 1998: la explosión

No se sabe si se debe a un cansancio general de los usuarios, pero 1998 es el año de los anuncios espectaculares. El mundo de la informática se da cuenta por fin de que Linux no es un juguete para estudiantes manitas. En enero de 1998, Netscape anuncia que su producto pasa a open source. De ahí saldrán Mozilla, Firefox y Thunderbird. Los institutos de formación añaden Linux a su catálogo. En julio de 1998, Oracle e Informix son portados a Linux. En septiembre, IBM lleva DB2, y Sybase hace lo mismo. Linus Torvalds aparece en la portada de la revista *Forbes*. KDE y Gnome llegan a la versión 1.0. En la bolsa, las cotizaciones suben, las empresas Linux nacen. Es el éxito.

En enero de 1999, llega Linux 2.2 y su imparable éxito hace reaccionar a Microsoft. Es David contra Goliat y sigue siendo así. Se hubiera podido pensar que la explosión de la burbuja de Internet en la bolsa en 2000 lo habría hecho fracasar todo. Pero ya se sabe que no. Linux no es un coloso de pies de barro. Sus pies son la comunidad, inquebrantable. El núcleo 2.4 sale el 4 de enero de 2001. El núcleo 2.6 sale el 18 de diciembre de 2003; el 3.0 en julio de 2011, para festejar los 20 años de Linux. En junio de 2014, el núcleo se encuentra en versión 3.15.

g. Hoy y mañana

Hoy Linux es famoso por ser un sistema operativo estable, robusto y eficiente. Se utiliza en más de un tercio de los servidores del mundo y dos tercios de los servidores web. Conquistó el mundo de la empresa, el mundo universitario y el mundo científico. El 98 % de los ordenadores del top 500 de los equipos más potentes del mundo tienen un sistema operativo basado en Linux.

Sobre todo, Linux ha sabido conservar su independencia, garantizada por la comunidad y el número de contribuyentes, frente a los gigantes de la informática. El próximo reto importante para Linux puede ser el entorno de escritorio y, por qué no, el hogar, en sustitución de Windows. Queda todavía camino por recorrer, pero muchos son los que ya han dado el primer paso.

Existen otros dominios donde Linux se impone y tiene un uso habitual. Incluso se puede afirmar que decenas de millones de personas lo utilizan habitualmente sin saberlo. Los distintos dispositivos de los proveedores de acceso a Internet funcionan mayoritariamente con Linux. La mayoría de los dispositivos multimedia también. Y, es sobre todo, en la informática móvil (teléfonos, tabletas, GPS, etc.) donde Linux se ha impuesto en proporciones gigantescas. Cualquier usuario que consulte las propiedades de su teléfono móvil o de su tableta Android podrá comprobar que es un núcleo Linux el que las hace funcionar. Teniendo en cuenta que más del 80 % de los smartphones funcionan con Android, ¡es el mundo entero el que usa Linux!

Un último dominio, muy de moda al ser muy sencillo, útil y práctico es la cloud. Muchos fabricantes ofrecen una plataforma de tipo cloud, reutilizando fundamentalmente las tecnologías de virtualización existentes. Linux es un actor principal en esta tecnología. Bien sea Amazon, Google u OVH, estos servicios emplean Linux y ofrecen soluciones articuladas entorno a Linux. Creamos una máquina virtual en la nube con un SO Linux en unos pocos clics. La plataforma Azure de Microsoft no es evidentemente gestionada por Linux, pero la mayoría de las máquinas que ejecutan en él son Linux…

¿Qué hardware es compatible con Linux?

1. La arquitectura

Linux es compatible con al menos cuatro arquitecturas físicas actuales:

- **x86** para los ordenadores cuyos procesadores son del tipo Intel (del 386 al Pentium 4) o AMD (Athlon, Duron, Sempron) de 32 bits. Esta versión funciona también en las máquinas con procesadores de 64 bits. De hecho, con la multiplicación de los procesadores de 64 bits, el empleo de Linux en esta arquitectura tiende a reducirse.
- **x86_64** para los ordenadores cuyos procesadores son del tipo Intel (Pentium 4 a partir de las series 600, Xeon, Dual Core/Quad Core, i3, i5, i7) o AMD (Athlon 64, Sempron 64, Opteron, Phenom, FX) de 64 bits. Esta versión no funciona en los procesadores de 32 bits. El uso de una versión de 64 bits de Linux en los equipos que lo soportan es muy recomendable, siendo las ventajas numerosas.
- **arm**: es un caso más particular, ya que esta familia de procesadores se usa fundamentalmente en sistemas embedidos, especialmente en dispositivos multimedia, en "cajas" multifunción, en routers, en lectores DVD, DivX y Blu-ray de salón, en GPS o en smartphones y tabletas. Esta arquitectura ha descendido tanto en 32 bits y 64 bits, mono o multiprocesador.
- **ppc** para los ordenadores cuyos procesadores son de tipo PowerPC, o sea, los antiguos ordenadores de la marca Apple. Esta versión no se instalará en las últimas máquinas Apple, basadas en un procesador Intel. Aún existen algunas distribuciones para esta arquitectura.

Hace algún tiempo algunos drivers o aplicaciones no estaban adaptados a los procesadores o distribuciones de 64 bits. Hoy en día esto ya no sucede, ya que casi todos los procesadores y los sistemas operativos son de 64 bits, incluyendo Windows (desde hace ya tiempo). Es aconsejable emplear una versión de 64 bits de Linux si el equipo lo soporta para permitir al sistema desplegar todo su potencial. Sin embargo, un sistema operativo de 32 bits funcionará perfectamente en una arquitectura Intel 64 bits pero con ciertas limitaciones: la memoria, por ejemplo, no sobrepasara los 4 GB aunque se cuente con una cantidad físicamente superior.

Configuración hardware básica

Linux soporta en teoría todos los tipos de procesadores desde la versión 386, y puede funcionar con sólo unos megabytes de memoria. La distribución Polux Linux funciona en un 386 con 4 MB de memoria. La distribución Damn Small Linux funciona con un 486, 16 MB de memoria y utiliza 50 MB de espacio en disco. Incluso encontramos distribuciones con uno o dos disquetes que se cargan en 2 MB de memoria. En máquinas un poco más recientes (de 32 a 128 MB de memoria) las distribuciones Toutou Linux o Puppy Linux lo hacen a las mil maravillas.

El núcleo 3.8 aporta la primera ruptura tecnológica importante desde el origen de Linux: los procesadores Intel i386 (80386 y asociados) no están soportados. Esto significa que si quiere instalar Linux en un equipo antiguo de finales de los años 80 o principios de los 90, deberá utilizar una versión anterior. El soporte comienza con los procesadores 80486, que llamamos también i486.

Sin embargo, no espere trabajar correctamente con una versión moderna de Linux y su entorno ofimático gráfico en estas condiciones pseudo-prehistóricas. Se deben respetar los requisitos siguientes:

- Un procesador (o más) de tipo Intel Pentium o superior o uno equivalente de la marca AMD. Atención: algunas distribuciones se compilan para funcionar únicamente a partir de Pentium (i586) o de Pentium Pro o II (i686) en adelante. En las distribuciones de 2010 y siguientes, los giga hertzios no son suficientes.
- Al menos 128 MB de memoria permiten arrancar un Linux básico, pero no son suficientes para

un manejo cómodo. Piense más bien en disponer de 1 GB para obtener el mejor rendimiento. Dado el precio de la memoria, esto no es un lujo. Parta del principio que cuanta más memoria, mejor. Los PC recientes con 2 GB o 4 GB de memoria son ideales. En el marco de una instalación mínima en modo texto, 64 MB bastan. Los 64 bits permiten explotar mucho más, no es raro encontrar servidores con 64, 128 o 256 GB de RAM. Si instala un Linux de 64 bits no olvide que la ocupación de las celdas de memoria (las palabras) se multiplica por dos. De esta forma, lo que entra en 1 GB puede ser muy justo.

- 500 MB de espacio en disco para una instalación mínima (sin interfaz gráfica y sólo con las herramientas básicas), pero de 2,5 GB a 4 GB para una instalación estándar, a la cual hay que añadir el espacio para los datos del usuario y la partición de intercambio.
- Una tarjeta gráfica (incluso antigua) compatible con la norma Vesa, que acepte una resolución de 1024 x 768 en 65 536 colores para el entorno gráfico. La resolución no tienen ninguna importancia si trabaja en modo texto.

Estos son los requisitos básicos. Si la frecuencia de reloj de su procesador es vital para la velocidad de ejecución de sus aplicaciones, ésta puede verse muy restringida por la falta de memoria o un disco duro demasiado lento. La cantidad de memoria física es un factor importante en el rendimiento. Cuanta más tengamos, mejor. Varios programas podrían funcionar paralelamente, la partición de intercambio no será necesaria y el sistema podrá utilizar más memoria caché para acelerar los accesos a los discos y periféricos. Si dispone de 512 MB o menos, considere ampliar a 1 GB o más. La diferencia es muy importante.

Las prestaciones globales siguen siendo aceptables en un Pentium II 300 con 256 MB para un uso ofimático o sencillo de Internet. Pero el sistema se resiente cuando varios programas se ejecutan al mismo tiempo. En un simple AMD Duron 800 con 512 MB, las prestaciones son excelentes para la mayoría de los usos habituales. En un servidor, según el uso un tamaño de memoria de 1GB puede bastar para hacer funcionar un buen servidor web basado en una arquitectura LAMP (*Linux Apache MySQL PHP*).

2. Compatibilidad del hardware

Antes de instalar Linux, compruebe si su hardware es compatible con Linux. Establezca una lista de los componentes de su ordenador y de sus diferentes periféricos. Los principiantes suelen alegar un problema con el soporte del hardware cuando falla la instalación. Una tarjeta gráfica, una impresora, un escáner determinados no funcionan correctamente o no del todo. Aunque Linux soporte la mayoría de los componentes de los ordenadores recientes, la compra de un ordenador moderno no es una garantía de buen funcionamiento.

En la lista, lo más importante no es la marca o el nombre del modelo comercial, sino el componente, el chip principal, llamado *chipset*, del producto. En el caso del Wi-Fi, poco importa que la tarjeta sea una Palmnet BZ46G. Pero si sabe que está construida a partir de un chip Centrino (Intel 2200 por ejemplo), entonces encontrará que funciona con Linux. Se deben evitar los productos de algunos fabricantes porque sus componentes no disponen de controladores que permitan utilizarlos.

La casi totalidad del hardware de impresión ofrecido por Hewlett-Packard o Samsung funciona perfectamente con Linux. Por el contrario, hay que huir de las impresoras de tinta Lexmark (cuidado: no es la calidad del producto lo que está en entredicho, sino su soporte en Linux).

Si ya está instalado, el sistema operativo Windows puede ayudarle a inventariar su hardware. En el Panel de control puede acceder a Sistema y a la pestaña **Hardware**. Ya sólo le falta comprobar mediante un motor de búsqueda o sitios web especializados si sus componentes son compatibles.

Los casos de incompatibilidades son cada vez más y más raros, a parte de cierto hardware específico o muy reciente. Muchos fabricantes, proporcionan lo necesario fundamentalmente para el buen funcionamiento de Linux en sus servidores, certificando la compatibilidad de sus modelos con ciertas distribuciones. Los servidores Proliant de HP son por ejemplo certificados para la mayoría de las distribuciones profesionales: Red Hat, Suse y Ubuntu LTS.

A menos de que disponga de una máquina muy antigua, todas las tarjetas gráficas funcionan. En todos los casos, Linux ofrece un controlador genérico llamado "vesa" que, si no ofrece las mejores prestaciones, al menos permite utilizar todas las tarjetas compatibles con este estándar de más de diez años. Algunos fabricantes ofrecen controladores muy eficaces. Las últimas tarjetas de los constructores Nvidia, AMD (ATI) e Intel soportan con drivers 3D que ofrecen las mismas prestaciones que en otros sistemas operativos. El sistema gráfico de Linux soporta por defecto un gran número de tarjetas, incluso con la aceleración 3D. Las mejores tarjetas gráficas han sido durante mucho tiempo las basadas en componentes NVIDIA mediante el controlador propietario e Intel mediante el controlador libre. Estos últimos años, gracias en gran medida a la ingeniería inversa en tarjetas NVIDIA (driver nouveau) y Radeon (driver Radeon), se han desarrollado drivers 3D libres. Su rendimiento ha alcanzado cotas comparables a los drivers propietarios.

Las tarjetas de sonido integradas en la placa base respetan un estándar de facto (AC97 o Intel HD) soportado por Linux. Las tarjetas de sonido integradas en la placa base son rara vez componentes de alta gama. Una simple tarjeta Live que cuesta menos de 30 euros es mucho más eficaz. Algunos modelos específicos de tarjetas de sonido pueden crear problemas.

El Wi-Fi (con USB, PCCard, PCI, PCIe) debería funcionar o con un controlador nativo para su hardware, o con la ayuda de una herramienta llamada Ndiswrapper que permite utilizar los drivers Windows para Linux. Según la versión que elija, puede ocurrir que no se le suministren pequeños componentes llamados firmwares, necesarios para la tarjeta Wi-Fi, y se deben adquirir por separado, ya sea desde el sistema de actualización o desde un soporte (el famoso add-on) adicional, o por el constructor de la tarjeta. Los chipsets de marca Ralink, Broadcom o Intel son generalmente compatibles. El Bluetooth es perfectamente reconocido y soportado.

Varios sitios disponen de bases de datos de hardware compatibles para informarle con precisión. Los motores de búsqueda siguen siendo su mejor fuente. A título orientativo, le presentamos una lista de los sitios que le ayudarán en sus búsquedas:

- Lista de compatibilidad Novell y OpenSUSE: http://en.opensuse.org/Hardware?LANG=en_UK
- Impresoras: http://www.linuxfoundation.org/collaborate/workgroups/openprinting
- Escáneres: http://sane-project.org/
- Todos los dispositivos: http://linux-drivers.org/
- Tarjetas de sonido: http://www.alsa-project.org/
- Tarjetas Wi-Fi: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/
- Módems internos o externos de tipo Windomem: http://linmodems.org/

A pesar de toda la buena voluntad del mundo, ocurre a veces que un dispositivo no funcione en absoluto con Linux. ¿Quién tiene la culpa? Los drivers de periféricos suelen ser escritos por desarrolladores que ni siquiera tienen acceso a las especificaciones del dispositivo y que lo hacen todo por ingeniería inversa, o sea, intentando reproducir el funcionamiento del periférico desde su resultado. Requiere mucho tiempo. Algunos fabricantes respetan las reglas. A falta de proporcionar un verdadero driver, publican para algunos desarrolladores una documentación técnica. Otros proporcionan ellos mismos un driver con código cerrado para una distribución dada o que debe adaptar usted mismo a cada nueva versión de Linux. En este caso, no se garantiza el driver (esto ocurrió, por ejemplo, con las tarjetas basadas en chipset gráfico Kyro II o powerVR). Por lo tanto, antes de criticar a Linux si no le funciona un dispositivo, critique primero al fabricante de este último.

Elegir una distribución

1. Debian

El proyecto Debian fue creado por Ian Murdock en 1993, en una época en la cual la idea misma de distribución Linux estaba todavía en pañales. El nombre de Debian proviene de Debra (la esposa de Murdock) e Ian. Durante mucho tiempo, Debian ha sido la única distribución entera y únicamente compuesta de programas libres y open source, lo que le sigue valiendo el nombre oficial de Debian GNU/Linux. Debian ha sido considerada durante algún tiempo la distribución Linux de referencia para la FSF. Las ventajas de Debian son muchas:

- un número gigantesco de paquetes (se cifran en miles),
- un programa de instalación de softaware llamado APT muy práctico y eficaz,
- una distribución 100 % open source,
- una estabilidad a prueba de bomba para un entorno de producción.
- muchos repositorios de programas

Estas ventajas conllevan también unos inconvenientes:

- paquetes a menudo antiguos,
- actualizaciones de la distribución irregulares y demasiado espaciadas,
- riesgos relacionados con la multiplicación de los paquetes y de los elementos dependientes,
- una instalación y una configuración complicadas.

Todos estos inconvenientes no tienen por qué ser defectos. ¿Tenemos que preferir una antigua versión exenta de bugs o la última versión de un producto cuya fiabilidad no ha sido totalmente probada?

Todos estos elementos convierten a Debian en una distribución ideal para los informáticos, los ingenieros y los administradores de sistemas y redes, los entornos de producción en empresa, los puristas del software libre, los aficionados iluminados que no temen tirarse a la piscina. En cuanto a los principiantes, pasarán un pequeño calvario si se proponen aprender de forma autodidacta.

En junio de 2014, Debian se encuentra en la versión 7.5.

2. Ubuntu

El multimillonario sudafricano Mark Shuttleworth, esencialmente conocido en mundo entero por haber sido uno de los primeros turistas espaciales, pero también por haber hecho fortuna

vendiendo su empresa Thawte, especializada en la seguridad, a Verisign, es un verdadero amante de la informática que ha contribuido como ningún otro al proyecto Debian. Para solventar algunos problemas de éste, creó la distribución Ubuntu Linux en 2005, con un presupuesto de 10 millones de dólares en contratación de desarrolladores. La palabra "Ubuntu" es un término del lenguaje africano bantú que significa "humanidad para los demás" o "soy lo que soy gracias a lo que somos todos". Este leitmotiv ejemplifica lo que la distribución se ha propuesto: ser un derivado de Debian, pero proporcionando los programas más recientes y orientados a la compatibilidad y la ergonomía. Ubuntu ofrece:

- una distribución que proviene de Debian,
- compatibilidad con los paquetes de Debian,
- un sistema de instalación muy simple,
- una edición cada 6 u 8 meses,
- un entorno gráfico agradable.

Esta distribución es ideal para los novatos y para los estudiantes. Se ha creado para ser la más fácil de instalar y de utilizar. Su éxito ha sido colosal y es la distribución más descargada e instalada desde hace varios años. Aunque a veces se le critica por sus decisiones, Ubuntu es muy innovadora. El número de repositorios y su paquetería son impresionantes. En las secciones de descarga de muchos proyectos libres, se ofrece casi siempre un paquete Ubuntu. Está disponible en versiones Desktop y Servidor, con un largo periodo de soporte.

Las versiones LTS (*Long Term Support*) aparecen cada dos años y disponen de un soporte extendido: cinco años a partir de la versión 12.04. Debido a su corto ciclo de aparición, las versiones LTS disponen de un nuevo núcleo reciente. En junio de 2014, la versión LTS y la versión actual coinciden: es la 14.04. LTS, soportada por la mayoría de los fabricantes y es una opción excelente para un servidor: se manifiesta muy estable y es por ejemplo la distribución elegida para trabajar con los servidores Amazon AWS, o los contenedores Docker.

3. Red Hat y Fedora

Logo Red Hat

Si existe una empresa comercial en el mundo de Linux que influyó y sigue marcando época, es la empresa Red Hat. Fundada en 1995 por Robert Young y Marc Ewing, edita la famosa distribución epónima cuya primera versión oficial data de 1994 (la empresa se fundó tras el lanzamiento de la distribución). El sistema de paquete RPM apareció con la versión 2.0. Fue tal la buena acogida de Red Hat que lleva casi diez años siendo la referencia. Cada versión era innovadora tanto en la integración de los programas como en su instalador (llamado "Anaconda") y sus herramientas de configuración.

En el año 2003, Red Hat publicó la versión 9.0, la última destinada oficialmente al gran público. Se confió las siguientes versiones al proyecto comunitario **Fedora**, que sigue sacando cada seis meses una nueva versión. Red Hat se concentra ahora en el mundo de la empresa con distribuciones comerciales llamadas **RHEL** (*Red Hat Enterprise Linux*) y cuyas características son:

- versiones profesionales destinadas a las empresas,
- soluciones que van del puesto de trabajo al servidor más potente,

- diferentes arquitecturas físicas,
- un soporte comercial,
- actualizaciones garantizadas durante siete años,
- 100 % libre.

Aunque es posible instalar una versión RHEL AS (*Advanced Server*) en un PC de oficina, se puede dudar del interés que tiene para una estación de trabajo o un principiante. Aunque libre (puede disponer de sus archivos fuente libremente), el coste del soporte es muy elevado. No obstante, si no le da miedo su instalación, la distribución **CentOS** (*Community Enterprise Operating System*) es una copia exacta y descargable de RHEL en la que se ha suprimido cualquier rastro de los nombres y logos de Red Hat. Red Hat o CentOS serán la elección del iniciado que desea aprender directamente con una distribución Linux utilizada en empresa.

La versión de RHEL en junio de 2014 es la 7.0. Esta dispone de un núcleo 3.10 (mientras que Ubuntu 14.04 LTS, publicada previamente, dispone de un núcleo 3.13).

Logo Fedora

En cuanto al proyecto Fedora, sigue un ciclo de desarrollo rápido y continúa destinado al gran público. Su instalación es sencilla. Sin embargo, al conjunto le falta un poco de coherencia. En versión 20 en junio de 2014, Fedora es la distribución ideal para todos aquellos que desean entrar un poco más en los pequeños detalles.

4. Mandriva (ex-Mandrake) y Mageia

Mandriva Linux (ex-Mandrake) es una distribución derivada de la distribución Red Hat y durante mucho tiempo completamente compatible con ella. Fue creada por Gaël Duval con el fin de integrar a la distribución el entorno gráfico de oficina KDE, al contrario que el de Red Hat, que integraba el entorno GNOME. Durante varios años, Mandrake fue la distribución estrella en fuerte competición con Red Hat. Mandrake era de hecho (y sigue siendo) más amena. Su proceso de instalación es un modelo a seguir y su utilización resulta de las más sencillas. Renombrada como Mandriva tras la adquisición por la empresa Connectiva, la distribución pierde influencia a partir de entonces. Las razones son múltiples, pero muy relacionadas con los avatares de la empresa Mandriva. Una gestión difícil y una mala orientación en los años 2000-2001 (el e-learning y la experiencia estadounidense de Start-up) estuvieron a punto de llevarla a su bancarrota una primera vez y, tras un procedimiento concursal del que consiguió salir, se encontró de nuevo con problemas poco después. La introducción en el mercado bursátil no dio los resultados esperados. Esta distribución es poco atractiva para los profesionales, que la consideran orientada al gran público. Sus soluciones, que satisfacen al gran público, siguen siendo consideradas de las mejores técnicamente, pero sufre algunos problemas de estabilidad.

Mandriva, no en vano, sigue innovando constantemente, en particular en la estación de trabajo nómada con versiones que pueden ser cargadas desde un USB. Esta distribución se beneficia del boca a boca entre los principiantes.

Logo de Mageia

Los últimos problemas de Mandriva han provocado la aparición de un derivado comunitario de la distribución, llamado Mageia. Se trata de una distribución de mucha calidad destinada al gran público. Esta distribución encontró un merecido éxito al encontrarse entre las 5 distribuciones más consultadas en 2014.

5. openSUSE

Se pronuncia *opensus*; **OpenSUSE** es una distribución de origen alemán del año 1992. El nombre de la empresa era de por sí un homenaje al famoso **Konrad Zuse**, el inventor de los ordenadores modernos.

La distribución está basada originalmente en Slackware. En 1996, SuSE se acerca a una distribución francesa llamada **Jurix**, creada por Florian La Roche, que se utiliza como base en lugar de Slackware. Este mismo año arranca el desarrollo de la herramienta YaST y sale la versión 4.2, en realidad totalmente nueva. Al mismo tiempo, SuSE emplea el nuevo gestor de paquetes de Red Hat, llamado RPM.

A principios de 1997, SuSE se embarca en la aventura americana al instalar nuevas oficinas en Oakland. Entre 1997 y 2003, la distribución SuSE no deja de ser mejorada para convertirse en una referencia en cuestión de sencillez de instalación, administración y uso.

Hoy en día, el futuro de la distribución está garantizado; la empresa Novell compró primero la empresa Ximian, especializada en el desarrollo open source de herramientas para Linux, entre las que destacan un completo escritorio Gnome, un sistema de mensajería llamado Evolution y un asistente de configuración llamado Red Carpet. Novell anuncia la compra de la empresa SuSE en enero de 2004. El desarrollo pasa ahora a ser comunitario, bajo el proyecto **openSUSE**. Cualquiera que lo desee puede participar en la mejora del producto. En contraprestación, Novell se compromete a proporcionar a la comunidad cada seis a ocho meses una versión estable, libre y gratuita. La reciente fragmentación de Novell en varias estructuras no ha disminuido la vida de la distribución, que sirve de base a la distribución SLES (*Suse Linux Enterprise Server*) de Novell, el gran competidor de Red Hat. En 2011, Attachmate compra Novell y separa SuSE, que se convierte en una entidad independiente.

Geeko, la mascota openSUSE

6. Las demás

Es imposible nombrar todas las distribuciones, dado las muchas que hay. Además de las grandes distribuciones que acabamos de reseñar, se debe hacer mención a otros nombres. La distribución**Slackware** es una de las más antiguas. Incluso se entregaba en un disquete. Durante

los primeros años de vida de Linux, la Slackware era la distribución de referencia para aprender a utilizar Linux. Es extremamente austera: su instalador se reduce a la mínima expresión y casi todas las configuraciones deben establecerse a mano, sin asistente. No cuenta con un gestor de paquetes (se trata de simples archivos de ficheros comprimidos). Todo ello hace que sea ideal para los manitas y los apasionados de Unix, pero no tanto para los principiantes.

La distribución **Gentoo** es muy peculiar. En vez de entregarle todos los programas ya listos para su uso, su instalador va a determinar exactamente con su ayuda la configuración de su máquina, y en particular su modelo de procesador. En función de ello, compilará (transformará el programa fuente en forma de lenguaje comprensible en lenguaje máquina) cada componente, que habrá seleccionado con todas las optimizaciones previstas para su hardware. Eso se llama una distribución fuente. El resultado puede ser interesante: las prestaciones de sus programas aumentan, siendo de media de un 10 al 20 % más rápida. ¡Pero a qué coste! La instalación no es ni de lejos fácil para los principiantes y, sobre todo, es muy larga: varias horas (incluso decenas de horas) según los programas que quiera y la potencia de su equipo.

Otra distribución sorprendente es la **LFS** (*Linux From Scratch*). No es precisamente una distribución, sino más bien una guía que le proporciona un método para construir su propia configuración. Paso a paso, le corresponde elegir sus diversos componentes y la configuración de su sistema. Así, se asegura de que obtiene exactamente la distribución que desea, ni más o menos. Pero también en este caso, los principiantes, e incluso de hecho los aficionados iluminados, lo evitan.

Al margen de todas estas distribuciones, encontramos numerosos derivados. **Aurox Linux** deriva de Red Hat. **PCLinuxonline** deriva de Mandriva. **Kubuntu** deriva de Ubuntu (o más bien es una distribución Ubuntu que integra el entorno de escritorio KDE), que deriva de Debian. **CentOS** deriva de RHEL, **Mint** deriva de Ubuntu, y así sucesivamente. Además, existen también las minidistribuciones que caben en un minicd o un pendrive, y es lo ideal para arreglar un ordenador.

Puede encontrar una lista exhaustiva de distribuciones Linux en distrowatch:http://distrowatch.com/

7. Los LiveCD

El LiveCD es una categoría sorprendente. Seguramente habrá probado Linux para ver cómo es o para comprobar si funciona correctamente con su hardware. En vez de instalarlo en su disco duro (si esta etapa le asusta, el capítulo El Shell y los comandos GNU de este libro le propone una guía paso a paso para instalar Linux), piense primero en probar Linux sin instalarlo. El LiveCD sirve principalmente para eso: es una instalación completa de Linux muy comprimida y que cabe en un solo CD o DVD (en este caso, se habla del liveDVD).

Utilizar un LiveCD es muy sencillo: inserte el CD o DVD en su lector y vuelva a encender el ordenador. Después de haber comprobado la configuración de su máquina (el setup del BIOS, véase las instrucciones de uso de su ordenador), su lector de CD o DVD debe ser el primer dispositivo en cargarse. Pasados unos segundos (o a veces minutos) de carga en memoria, aparece el escritorio y todos los programas más conocidos. No se instala nada en su disco duro. El LiveCD más famoso actualmente es **Knoppix**. Está basado en una distribución Debian y, además, cuenta con un mecanismo para copiarlo en el disco duro. Cada nueva versión de la mayoría de las distribuciones (openSUSE, Fedora, Ubuntu, Mageia, etc.) llega con un LiveCD para probar las últimas novedades sin instalarla.

La mayoría de los LiveCD, o las mismas distribuciones, pueden ser arrancadas desde una llave USB.

Una última categoría es la de los discos de recuperación. Ultimate Boot CD o System Rescue CD ofrecen varias herramientas que usan Linux para reparar el sistema. Cabe destacar que con estas herramientas es muy fácil desde Linux averiguar una contraseña Windows si la ha perdido.

Obtener ayuda

1. La ayuda propia de los comandos

No es posible conocer de memoria todos los parámetros y argumentos de un comando. Por fortuna, Linux propone como mínimo dos mecanismos para conocer la sintaxis de un comando. La mayor parte del tiempo, el parámetro `--help` muestra la ayuda incluida directamente dentro del programa al que se llama. A veces la ayuda es suficiente para encontrar lo que está buscando. Es lo que sucede con el comando **date**, cuya salida está voluntariamente cortada aquí porque, si no, necesitaría dos páginas.

```
$ date --help
Uso: fecha [OPTION]... [+FORMAT]
    o: fecha [-u|--utc|--universal] [MMDDhhmm[[CC]AA][.ss]]
Mostrar la fecha corriente según el FORMAT especificado o
inicializar la fecha del sistema.

  -d, --date=CADENA         mostrar la fecha según la descripción dada
                            por la CADENA,
                            excluyendo la palabra reservada "now"
  -f, --file=ARCHIVO        idéntico a --date para cada línea del
                            ARCHIVO de fechas
  -r, --reference=FILE      display the last modification time of FILE
  -R, --rfc-2822            output date and time in RFC 2822 format.
...
```

Puede ocurrir que la ayuda sea demasiado concisa o le falten explicaciones, o bien que esté totalmente ausente. En este caso, se considera `--help` como un parámetro no válido y puede que obtenga un mensaje de error o una línea de información:

```
$ cal --help
cal: opción inválida -- -
uso: cal [-13smjyV] [[mes] año]
```

La última línea no explica la sintaxis de los parámetros.

2. La ayuda interna del shell

Los comandos internos no aceptan el parámetro `--help`, pero para estos comandos el intérprete de comandos propone un comando **help**. Si se utiliza de forma aislada, proporciona la lista de los comandos internos. Si le proporciona como parámetro el nombre de un comando interno, aparece la ayuda correspondiente. De esta manera, puede aprender que pwd admite dos parámetros opcionales.

```
$ help pwd
pwd: pwd [-LP]
    Print the current working directory.  With the -P option, pwd
prints the physical directory, without any symbolic links; the -L
option makes pwd follow symbolic links.
```

3. El manual en línea

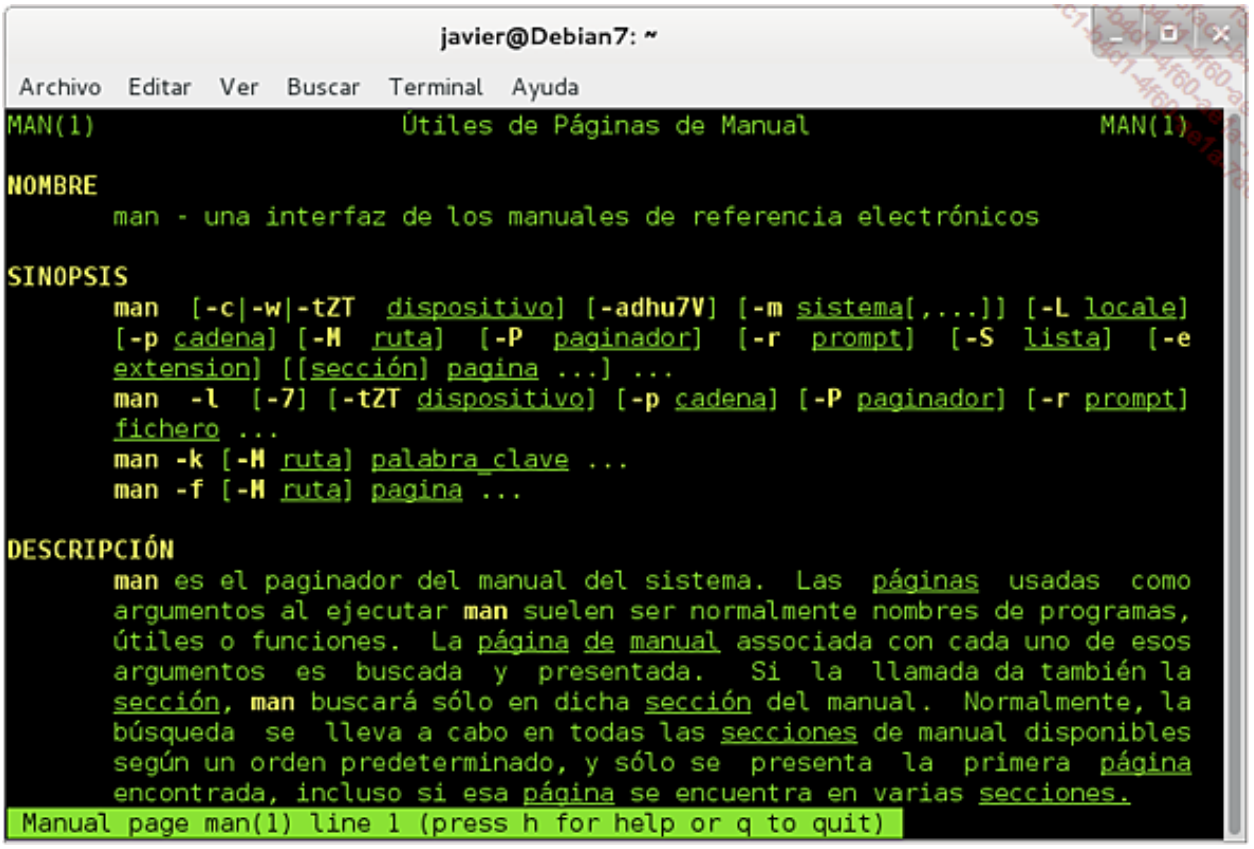
a. Acceso

Cuando los dos mecanismos de ayuda anteriores son insuficientes, es probable que la ayuda buscada se encuentre en el manual de Unix. Esta manual es estándar en todos los Unix, incluso Linux, y no importa cuál sea el shell, ya que se trata de un comando externo.

Al manual se accede con el comando **man**. Puede hacer una sencilla prueba con el comando **date**:`$ man date`

Aparecen las instrucciones del comando del parámetro.

b. Estructura de una página



El manual en línea

Una página de manual se compone de varias secciones, entre las cuales están las siguientes, aunque no están todas obligatoriamente:

- **Nombre**: nombre y papel del comando.
- **Sinopsis**: sintaxis general, parámetros y argumentos aceptados.
- **Descripción**: instrucciones detalladas del funcionamiento del comando y de los argumentos principales.
- **Opciones**: descripción detallada de cada parámetro posible, en general en forma de lista.
- **Ejemplos**: el manual puede proporcionar ejemplos concretos de uso del comando.
- **Entorno**: el comando puede funcionar de manera diferente dependiendo de los valores que adopten algunas de las variables del shell.
- **Conformidad**: el comando se ajusta a unas recomendaciones o normas (por ejemplo, POSIX).
- **Errores (bugs)**: el comando puede a veces funcionar mal en casos puntuales que se pueden enumerar en este sitio.
- **Diagnóstic/retorno**: el comando, según su resultado, puede devolver códigos de errores significativos cuyo valor permite determinar el tipo de problema (archivo con argumento ausente, etc.).
- **Ver también**: lista de los comandos relacionados con el programa que pueden interesar al usuario.

c. Navegación

Se navega muy fácilmente por la ayuda:

- La barra de espacio desplaza una página completa.
- La tecla [Intro] desplaza línea por línea.
- Las teclas [Arriba] y [Abajo] desplazan una línea arriba o abajo.
- Las teclas [Re Pág] y [Av Pág] desplazan media página arriba o abajo.
- Las teclas [Principio] y [Fin] hacen exactamente lo que se espera de ellas.
- La tecla / permite una búsqueda. /toto busca toto. En este caso, la tecla n busca la coincidencia siguiente, mientras que [Mayús] **n** (N) busca la anterior.
- La tecla Q sale de la ayuda y vuelve al shell.

d. Las secciones

El manual de Linux no sólo hace referencia a los comandos clásicos. Es un manual mucho más completo. Los comandos simples, los de administración, los archivos de configuración, los periféricos, las llamadas al sistema, las funciones de programación de varios lenguajes y mucho más pueden tener reflejo en estos manuales. Por eso, el manual se compone de varias secciones.

Sección	Contenido
1	Instrucciones ejecutables o comandos del shell
2	Llamadas del sistema (API del núcleo...)
3	Llamadas de las librerías (funciones C...)
4	Archivos especiales (contenido de /dev como sd, hd, pts, etc.)
5	Formato de los archivos (/etc/passwd, /etc/hosts, etc.)
6	Juegos, salvapantallas, programas varios, etc.
7	Varios, comandos no estándares que no encuentran sitio en otra parte
8	Comandos de administración del sistema Linux
9	Subprogramas del núcleo (a menudo vacío)

Puede ocurrir que a veces la llamada al manual para un comando no devuelva la página del manual correspondiente. Lo que pasa es que man busca por defecto la primera coincidencia en el orden de las secciones. Si busca ayuda relativa al formato del archivo de las contraseñas, llegará primero a la ayuda del comando **passwd**. Mire el encabezamiento de la página. El número de la sección está indicado justo detrás del nombre del comando, entre paréntesis. El comando **man** ha encontrado una coincidencia de **passwd** en la sección 1 y muestra la página del manual asociada.

```
$ man passwd
PASSWD(1)                Manual del usuario Linux
NOMBRE
       passwd -  actualizar las marcas de autenticación de un usuario.
...
```

Puede pedirle a man que busque el manual correspondiente en una sección específica indicando su número justo antes del nombre del comando. Para acceder al manual del archivo **passwd**, haga como a continuación:

```
$ man 5 passwd
PASSWD(5)              Manual del administrador Linux
NOMBRE
```

```
       passwd - Archivo de contraseñas.
...
```

e. Buscar por correspondencia

Si duda del comando que debe utilizar o no recuerda su nombre, o incluso si quiere conocer todos los comandos relacionados con una palabra, entonces utilice el parámetro `-k` de man:

```
$ man -k passwd
/etc/rpasswd.conf (5) [rpasswd.conf] - configuration file for remote
password update client
chpasswd (8)         - change user passwords in batch
Crypt::SmbHash (3pm) - Perl-only implementation of lanman and nt md4
hash functions, for use in Samba style smbpasswd entries
fgetpwent_r (3)      - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
gpasswd (1)          - change group password
ldappasswd (1)       - change the password of an LDAP entry
lppasswd (1)         - add, change, or delete digest passwords.
makepasswd (1)       - generate and/or encrypt passwords
mkpasswd (1)         - Overfeatured front end to crypt(3)
pam_localuser (8)    - require users to be listed in /etc/passwd
pam_rpasswd (8)      - PAM module to change remote password
passwd (1)           - change user password
passwd (1ssl)        - compute password hashes
passwd (5)           - password file
passwd2des (3)       - RFS password encryption
rpasswd (1)          - change user password on remote server
rpasswd.conf (5)     - configuration file for remote password update client
rpasswdd (8)         - remote password update daemon
saslpasswd2 (8)      - set a user's sasl password
smbpasswd (5)        - The Samba encrypted password file
smbpasswd (8)        - change a user's SMB password
vncpasswd (1)        - set passwords for VNC server
yppasswd (1)         - change your password in the NIS database
```

4. Las páginas info

El commando `info` permite leer las páginas en formato info. Esto es similar al manual, pero las páginas están estructuradas, ofrecen vínculos, un índice, una jerarquía o menús. Las páginas info pueden llamarse entre ellas. Ellas ofrecen frecuentemente una navegación amigable. Están escritas empleando las herramientas textinfo.

```
$ info date
```

El símbolo "?" muestra todos los comandos disponibles en info. La "q" permite salir.

Para obtener explicación de los parámetros de uno de los comandos, emplee la siguiente sintáxis:

```
$ info -show-options date
```

Finalmente, para encontrar todas las paginas info vinculadas a una palabra clave, haga lo siguiente:

```
$ info -apropos date
```

La mayor ventaja de info es la de ofrecer frecuentemente ejemplos muy detallados de los comandos de los lenguajes. Por ejemplo, escriba:

```
$ info sed
```

Con las teclas de dirección, descienda a la línea "ejemplos ::" y pulse en [Entrar]. Este vínculo le llevará directamente a la sección de los ejemplos de sed.

5. Buscar ayuda en Internet

Como ya se ha indicado al principio de este capítulo, hay toda una comunidad alrededor de Linux y del software libre, y los editores de distribuciones proporcionan la documentación y el soporte. Por eso, se dispone de muchos medios para obtener ayuda, en particular en Internet:

- la documentación del editor,
- los sitios comunitarios (FAQ, foros),
- los grupos de noticias,
- el proyecto de documentación libre (HOWTOs),
- etc.

Piense primero en la documentación de los editores:

- Red Hat: http://www.redhat.com/support
- Debian: http://www.debian.org/doc/
- openSUSE: http://en.opensuse.org/Documentation
- Ubuntu: https://help.ubuntu.com/
- Mandriva: https://doc.openmandriva.org
- Mageia: http://www.mageia.org/es/support/
- Fedora: http://docs.fedoraproject.org

En cada uno de estos sitios, encontrará también probablemente:

- una base de conocimiento,
- un Wiki,
- un foro,
- informes de bugs.

Resulta imposible relacionar todos los sitios de la comunidad; pero a continuación encontrará algunos:

- LinuxEs: http://www.linux.es/
- Freshmeat: http://freshmeat.net
- Slashdot: http://slashdot.org
- Foro Linux: http://www.linuxespanol.com
- Foro Fedora: http://fedora-es.com/
- Foro Mandriva: http://forum.mandriva.com/index.php?op=Es
- Foro Debian: http://www.esdebian.org/foro
- Foro Ubuntu: http://ubuntu-es.org/index.php?q=forum
- Foro openSUSE: Alionet: http://es.opensuse.org/punto_de_encuentro

Entre los sitios de documentación:

- Lea Linux: http://lea-linux.org
- The Linux Documentation Project: http://tldp.org
- LinuxDocs: http://linuxdocs.org

Instalar una Debian

1. Soporte de instalación

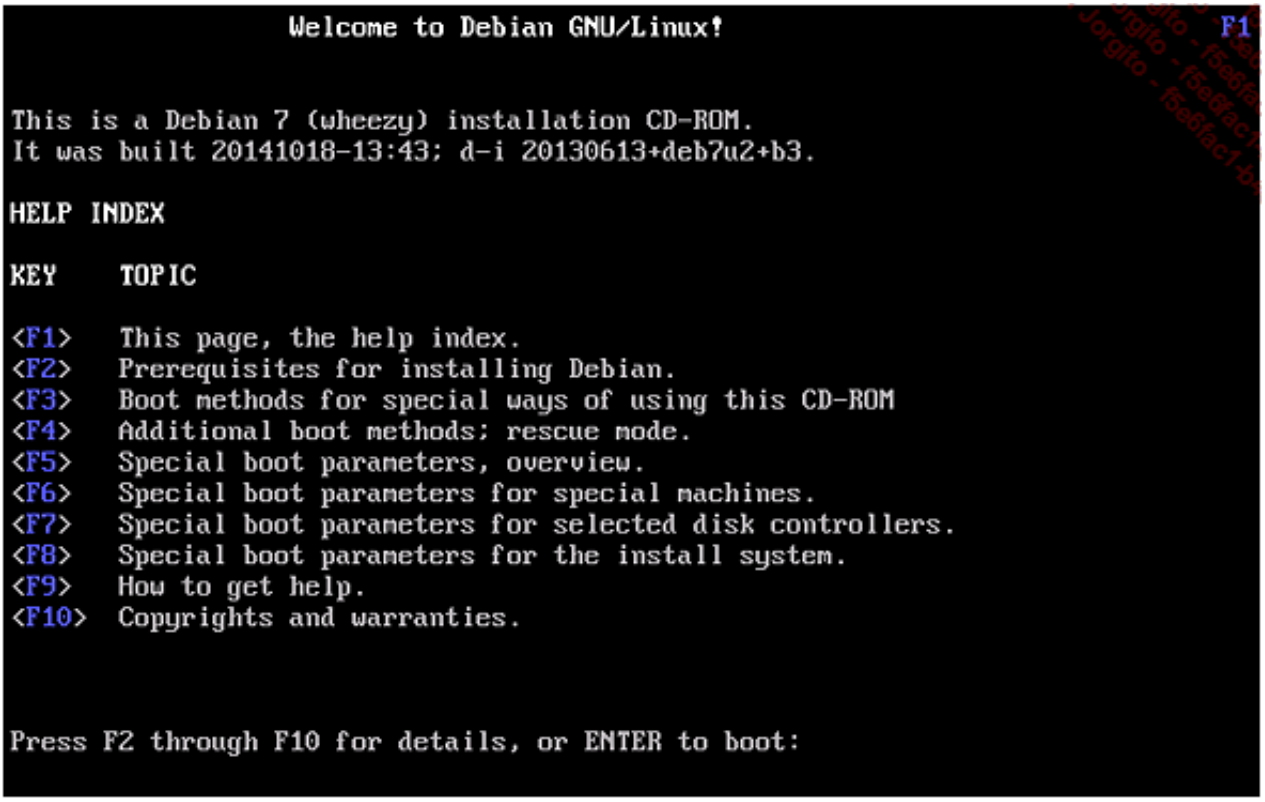
A continuación se explica, paso a paso, cómo instalar una Debian. Se trata de la última versión estable, o sea, la 7.5 Wheezy, en vigor durante la redacción de este libro. La instalación la haremos en modo texto porque representa el mejor método para Debian (existe un instalador gráfico poco utilizado). Si desea efectuar una instalación remota, debe obtener la imagen ISO (en el libro usamos la versión de 64 bits) correspondiente mediante la red (Internet), llamada netinst y disponible en el sitio de Debian: http://cdimage.debian.org/debian-cd/

Guarde esta imagen en un CD-ROM o en un pendrive. Para las necesidades de este libro, se instaló Debian en una máquina virtual VirtualBox 4.3.12, que es un producto gratuito y del que existe una versión libre.

2. Boot en el soporte

Boot de soporte Debian

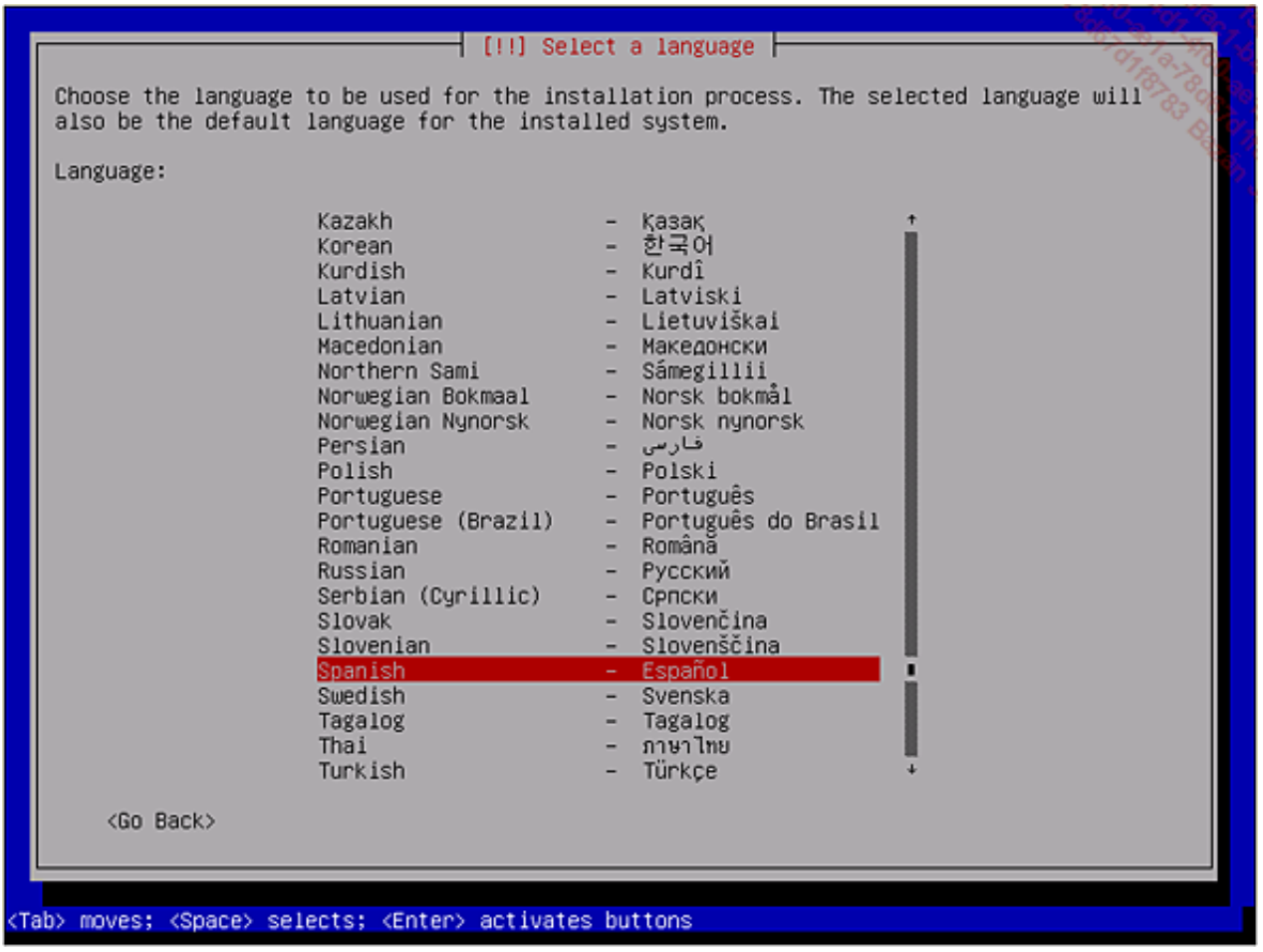
Configure su ordenador para que se inicie en modo instalación. En el momento de ejecutarse, usted tendrá acceso a una línea de comandos que permite dar paso a la instalación presionando [Entrar]. Las teclas de flecha permiten desplazarse por los menús. Si se desplaza hasta **Help** y escribe [Entrar], utilice las teclas de [F1] a [F10] para acceder a pantallas de ayuda adicionales. Considere las opciones en línea de comandos según su máquina (de [F5] a [F10]), ya que en algunos casos puntuales puede ser necesario modificar los valores que se pasan al núcleo de Linux para un buen funcionamiento.



Opciones de ayuda de boot

Presione la tecla [Entrar] para iniciar la instalación. Comprobará que el propio instalador funciona bajo Linux y que le muestra todas las líneas de inicio del núcleo de Linux. Finalmente, aparece la primera pantalla del instalador.

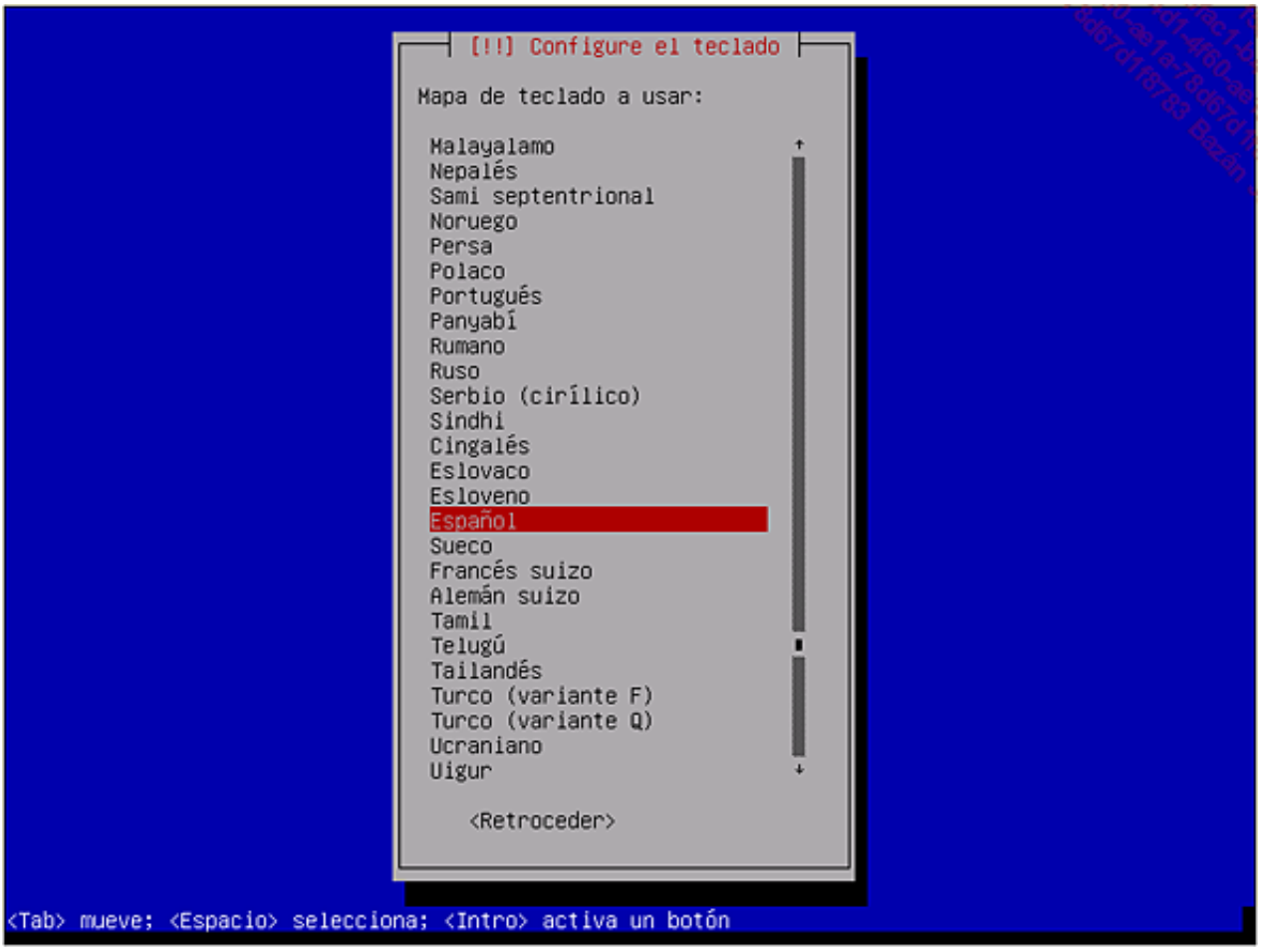
3. Elección de los idiomas y países



Selección de idioma por defecto

Las siguientes tres pantallas le permiten elegir:

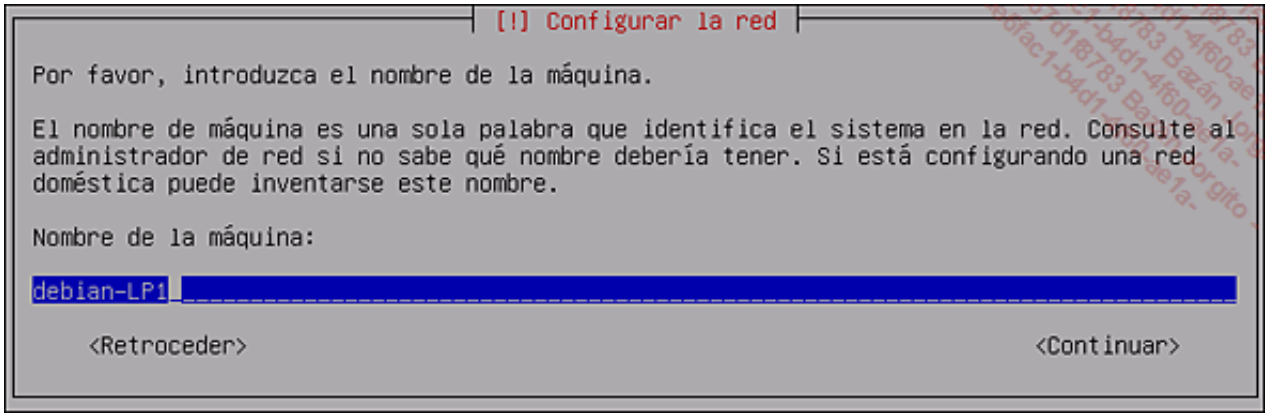
- El idioma utilizado por el proceso de instalación. Navegue con las flechas, pulse [Entrar] para seguir. A continuación, se utiliza el español.
- Según el idioma inicial elegido, Debian le ofrece un país. Según el país, Debian configura los formatos de fecha, hora, alfabeto, formatos numéricos y monetarios, etc.
- Para terminar, elija su tipo de teclado. Para España, es el es-latin9.



Selección del tipo de teclado

4. Anfitriones y usuarios

Las etapas siguientes consisten en introducir un nombre de anfitrión (el nombre de la máquina en la red) y el nombre del dominio. Si su equipo no pertenece a ningún dominio, deje el campo vacío.



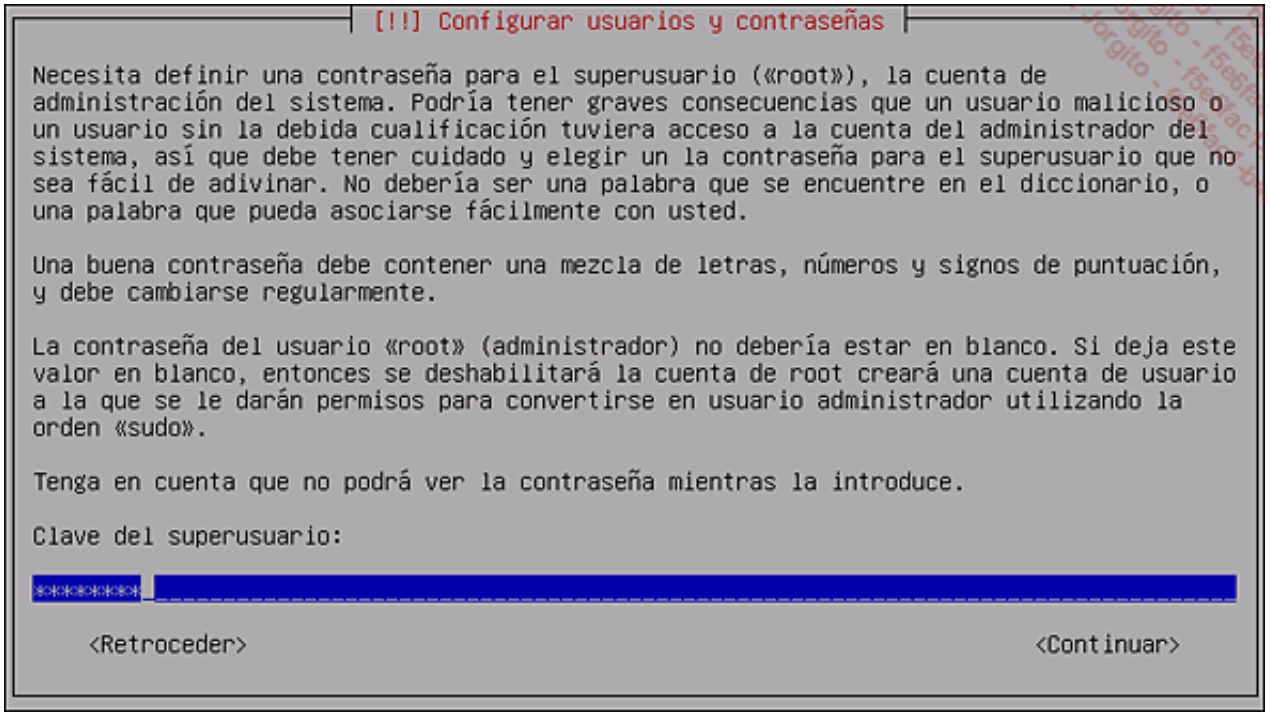
Nombre de host del equipo

5. Cuentas root y usuarios

Ahora debe introducir la contraseña del administrador root de la máquina. Se la pedirá dos veces, por confirmación. ¡No la pierda! Aunque existan algunos métodos para reinicializar la contraseña, o*craquearla* mediante una herramienta especializada, no hay ninguna manera de recuperar la

contraseña original. Debian incluso le avisa de que puede rechazar una contraseña si es demasiado simple.

Luego debe crear un usuario "normal". Podrá crear otros usuarios una vez terminada la instalación. Deberá introducir el nombre completo de la persona y Debian le propondrá un login. Introduzca luego la contraseña.



Introduzca la contraseña de root

6. Particionado de los discos

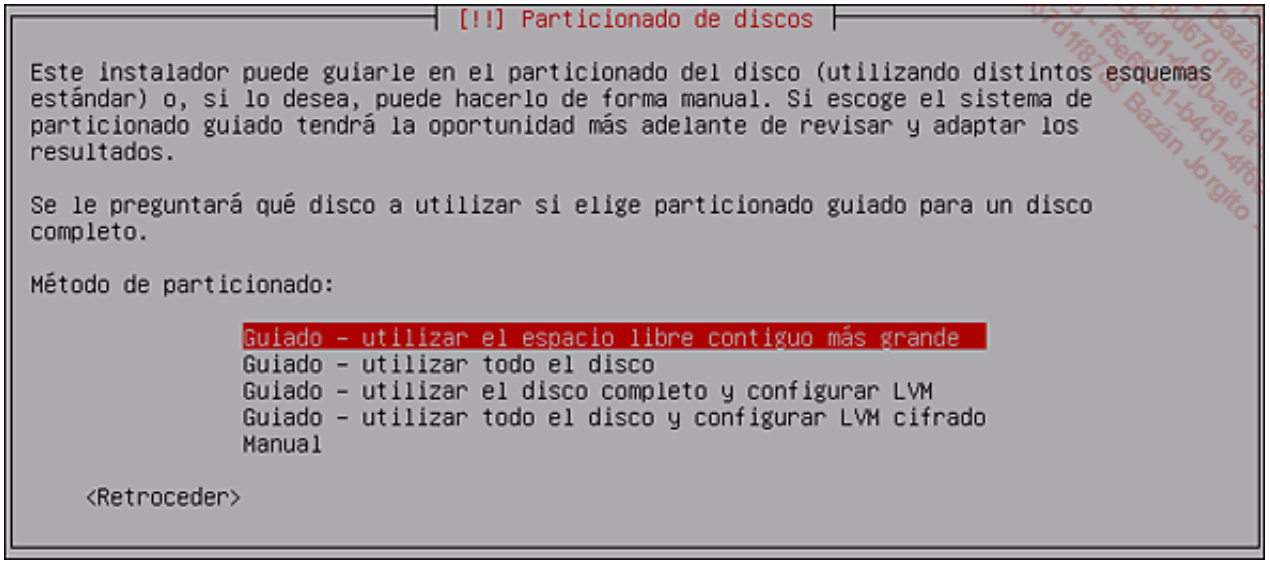
Con el fin de simplificar, diremos que dispone de tres métodos principales para particionar sus discos:

- Un método asistido (o automático) al utilizar la partición clásica (véase para ello el capítulo Los discos y el sistema de archivos).
- Un método asistido que ofrece el LVM (*Logical Volume Manager*).
- Un método manual.

El método asistido clásico, en el caso de una nueva instalación, da buenos resultados. Si vuelve a instalar una máquina, o instala Debian en una máquina que dispone ya de particiones que contienen las carpetas personales, por ejemplo, pase a un particionado personalizado.

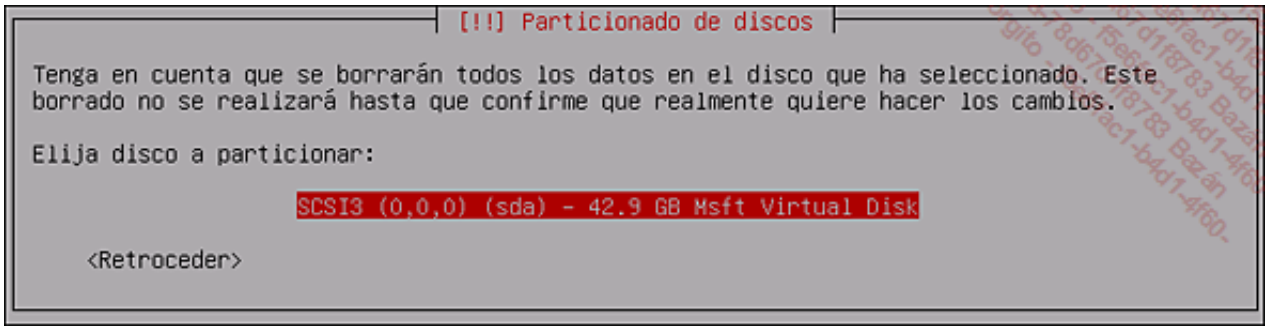
El LVM funciona agrupando discos físicos o particiones (llamados volúmenes físicos) en un solo espacio grande (llamado grupo de volúmenes), en el cual puede seccionar espacios lógicos a voluntad (llamados volúmenes lógicos), ampliarlos, reducirlos, etc.

Sin embargo, debería considerar la solución LVM en el contexto de un servidor de empresa o si piensa agregar luego discos en su máquina para añadir espacio de almacenamiento. El LVM aporta mucha flexibilidad.



Etapa de particionado

Si ha elegido el primer método, accede a la pantalla siguiente. Para llevar a cabo las tareas propuestas en este libro, se ha creado un espacio de 42 GB (más o menos) bajo Msft Virtual Disk como primer disco SCSI. Este servirá para la instalación.

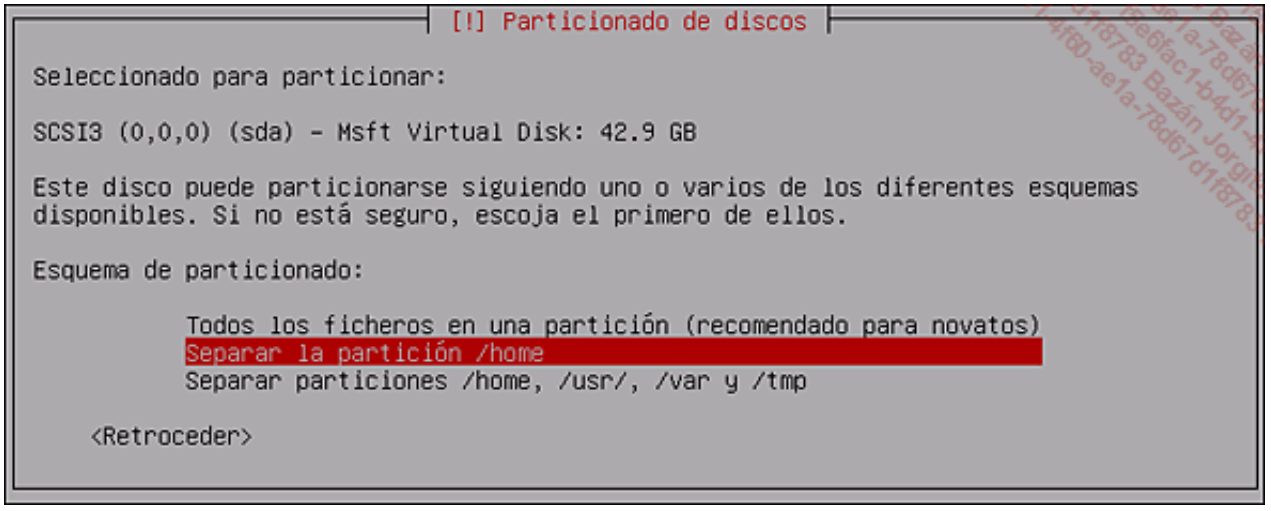


Selección del disco de instalación

La etapa siguiente consiste en elegir el esquema de particionado:

- O una sola partición grande en la cual lo pone todo (sistema, programas, datos). En tal caso, sólo necesita la partición raíz /.
- O dos particiones: una partición raíz que contendrá el sistema y todos sus componentes (programas, parámetros del sistema, etc.), y una partición que contendrá los datos de los usuarios. Para una estación de trabajo o un PC personal (en casa), es el método más pertinente: permite instalar de nuevo fácilmente otro sistema (actualización o reinstalación completa) sin comprometer los datos personales. La nueva distribución instalada de esta manera podrá utilizar de nuevo la partición montada en /home y, por lo tanto, recuperar los datos.
- El tercer método consiste en crear cinco particiones diferentes: la raíz /, los datos personales /home, los componentes de los usuarios (programas, librerías, datos compartidos asociados, etc.) /usr, el contenido variable /var y los archivos temporales /tmp. Esta opción es perfectamente apropiada para un servidor. Los mails, información DHCP, sitios web, etc., a menudo están almacenados en /var. Las actualizaciones de los distintos programas (servicios) están en /usr, etc. Este esquema de particionado es casi perfecto si ha seleccionado una instalación de tipo LVM: todo es independiente. De este modo, resulta mucho más sencillo cambiar el disco, migrar los datos, extender el tamaño de los volúmenes, etc., sin comprometer el resto.

La mejor opción para la instalación de prueba es la segunda, **separar la partición /home**. Las siguientes operaciones se basan en esta.



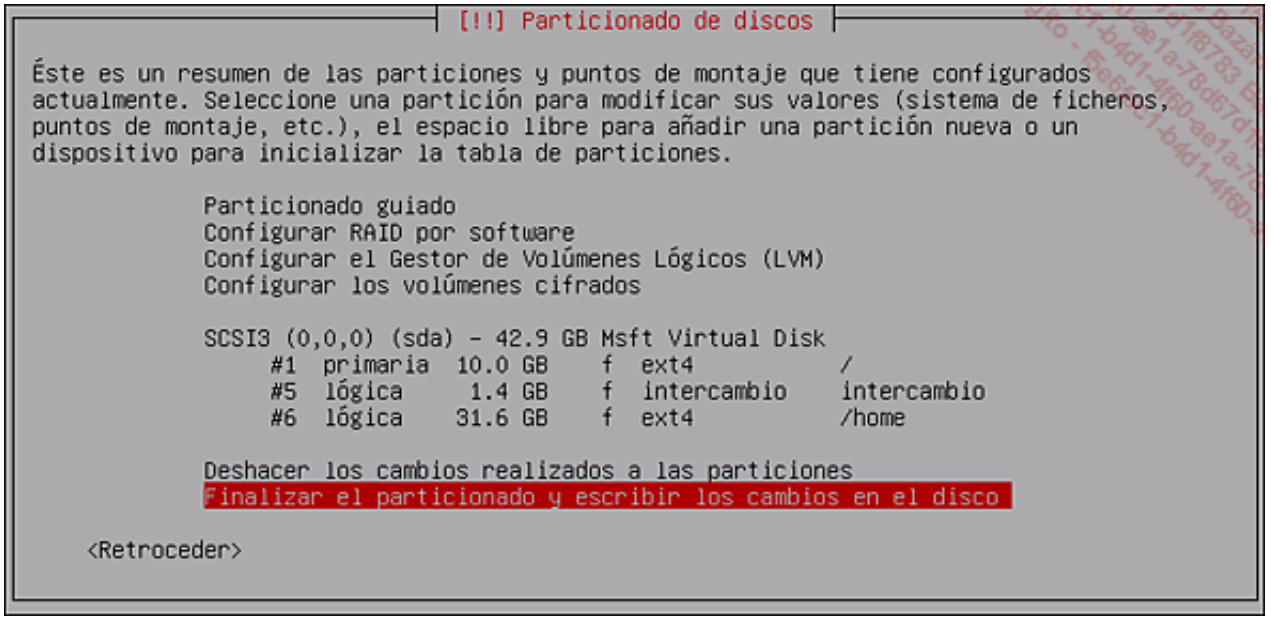
Selección del esquema de particionado

Al tratarse de un método semiautomatizado, Debian le indica sus opciones para cada una de las particiones. Puede constatar la presencia de tres particiones en lugar de las dos propuestas. Debian ha analizado el tamaño de la máquina y propone la creación de una zona de swap que corresponde a la mejor opción posible.

No se sorprenda tampoco por la cantidad de particiones. Muy a menudo, la única partición primaria es la raíz /, mientras que todas las otras son particiones lógicas dentro de una partición extendida, y Linux numera las primarias de 1 a 4 (una partición extendida es una partición primaria), mientras que la numeración de las particiones lógicas empieza con el 5. Para una mejor comprensión, véase el capítulo Los discos y el sistema de archivos.

Si este esquema de particionado le conviene, valide. La pantalla siguiente muestra un resumen que debe validar de nuevo.

¡Cuidado! El particionado va seguido de la escritura de los sistemas de archivos en las particiones correspondientes. Esta operación es idéntica al particionado y formateado en Windows y supone la destrucción de la información de la partición.



Confirmación definitiva de particionado

Una vez validados los cambios, una barra de progreso le informa del estado del particionado y de la escritura de los nuevos sistemas de archivos. Debian montará después estos sistemas para la

instalación.

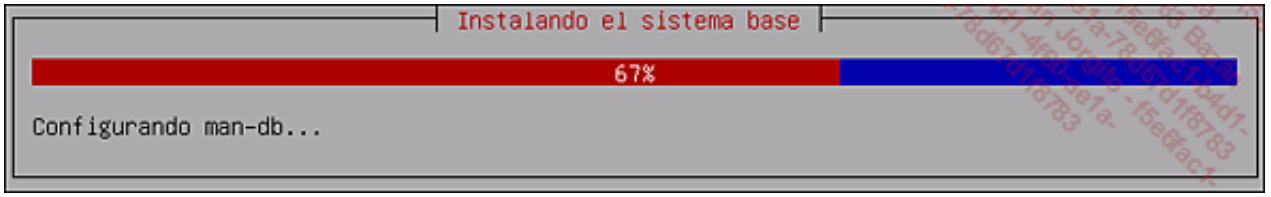
7. Instalación

El soporte de instalación elegido contiene lo necesario para instalar un sistema básico en su disco. En esta etapa, automática, se instalarán todos los paquetes básicos (núcleo, librerías, shell, etc). A continuación, si no desea instalar paquetes, tendrá un sistema minimalista pero funcional. En este momento y en la etapa de la elección de un servidor réplica del repositorio de paquetes se produce la configuración de red.

Si el instalador no ha conseguido configurar la tarjeta de red con DHCP, solicitará la introducción de los datos básicos: dirección IP, máscara de subred, puerta de enlace predeterminada y DNS.

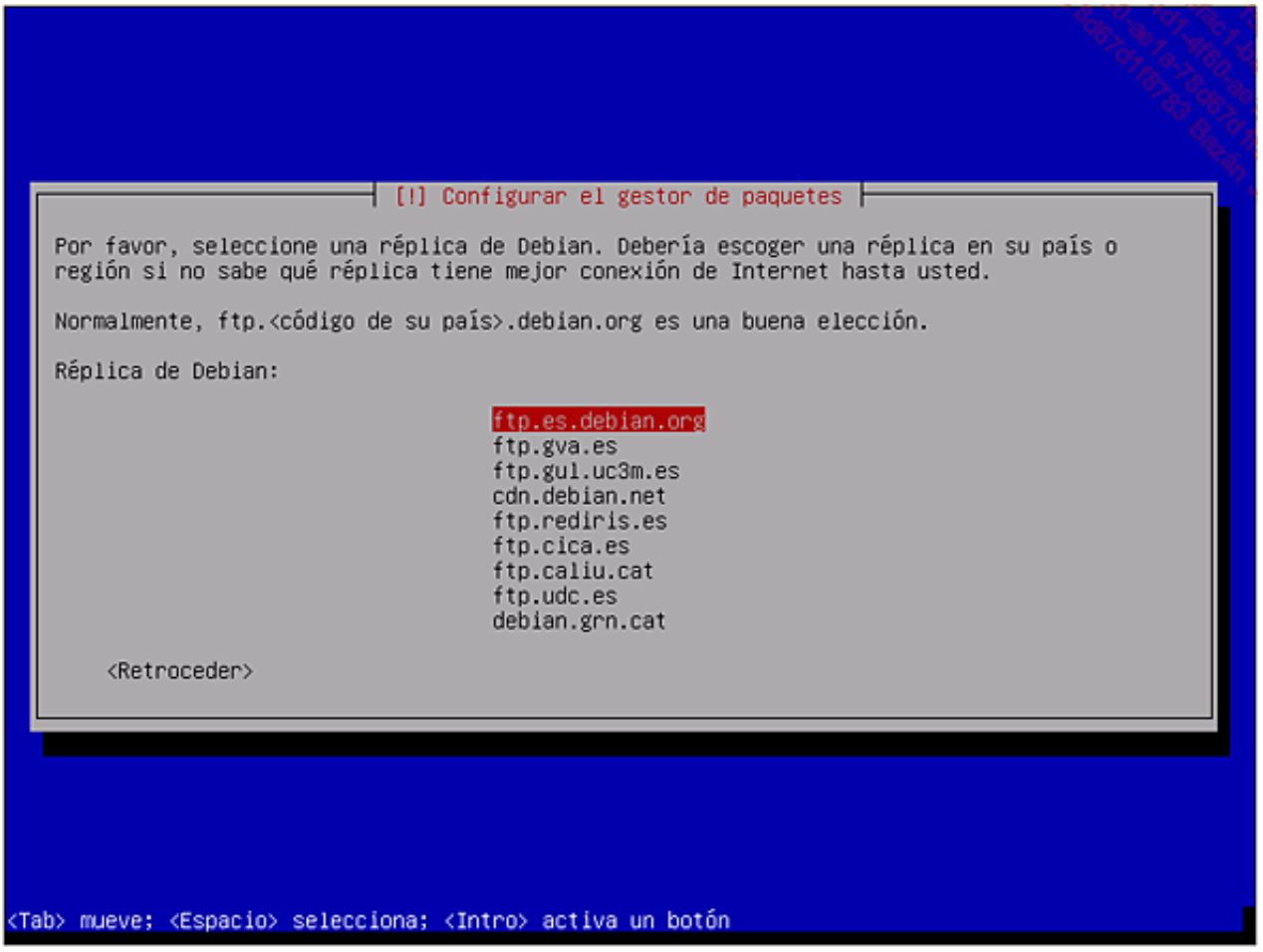
La instalación se desarrolla en varias etapas:

- Instalación de los elementos básicos. Ya se trate de elementos descargados o contenidos en el soporte de instalación, se vuelven a copiar los elementos básicos en su disco. Se trata de los paquetes principales.



Instalación del sistema base

- Luego elija una réplica (mirror) de Debian: es el lugar (en Internet) donde está depositado todo el software (repositorios) Debian. Seleccione en la medida de lo posible un mirror en su país o cercano a él. Por ejemplo en España el repositorio ftp.es.debian.org es el ideal. Si no utiliza repositorios, Debian usa únicamente el contenido del soporte de instalación. Si se trata de un DVD, no hay problemas, pero si utiliza, como aquí, un CD de instalación en red, sólo dispone de la base y nada más. Las réplicas contienen también las actualizaciones, en particular de seguridad, aparecidas entre el momento en el cual salió el software y el momento de la instalación. Se aconseja encarecidamente que configure uno de estos repositorios aunque disponga de un juego de soportes. En caso contrario, tendrá instalado un sistema básico. Puede elegir antes de esta etapa emplear un servidor intermediario, que llamamos también proxy. En las empresas es frecuente que el proxy sea quien filtra el acceso a Internet. Si es un particular, deje los campos vacíos.



Selección del espejo de repositorio Debian

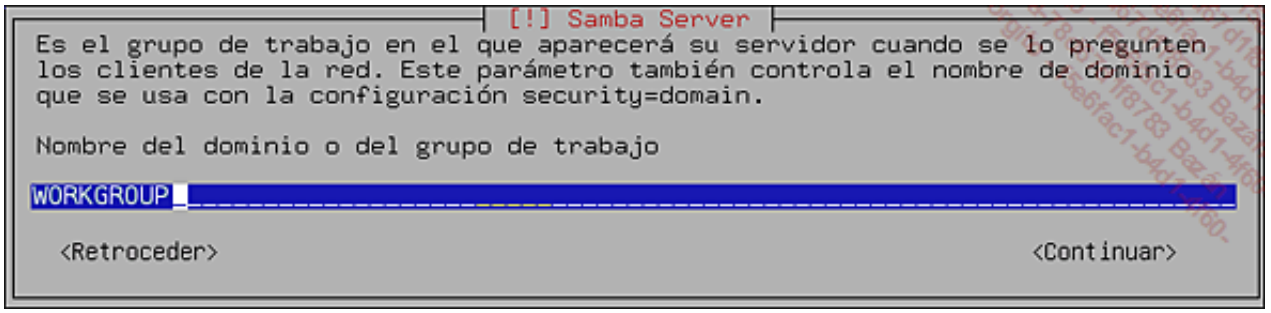
- Más adelante debe elegir entre uno o varios conjuntos de programas predefinidos. Estos conjuntos se agrupan por temas: entornos de escritorio (kde, gnome, xfce, etc.), los diferentes servidores (archivos, impresión, dns, correo, SQL, etc.). Para el marco de esta instalación, marque las opciones indicadas en la captura siguiente, sabiendo que el **SSH server** y **Utilidades estándar del sistema** son requisitos previos básicos.

Selección de los elementos predefinidos a instalar

Finalizada esta etapa de configuraciones, Debian comienza su instalación.

8. Configuración de los paquetes

Durante la instalación, quizás deba configurar algunos paquetes. Es el caso de, por ejemplo, el servidor Samba (particiones para Windows), donde se le pedirá el tipo servidor, el dominio, etc. Sucedería lo mismo con la configuración del servidor gráfico. No es posible darle una lista de cada una de las etapas de configuración de estos paquetes. Sin embargo, en caso de errores en la configuración es posible volver a reconfigurar el paquete correspondiente (aparece su nombre arriba; en el ejemplo se trata de xserver-xorg) usando el comando dpkg-reconfigure (p. ej.: **dpkg-reconfigure** xserver-xorg).



Configuración de Samba durante su instalación

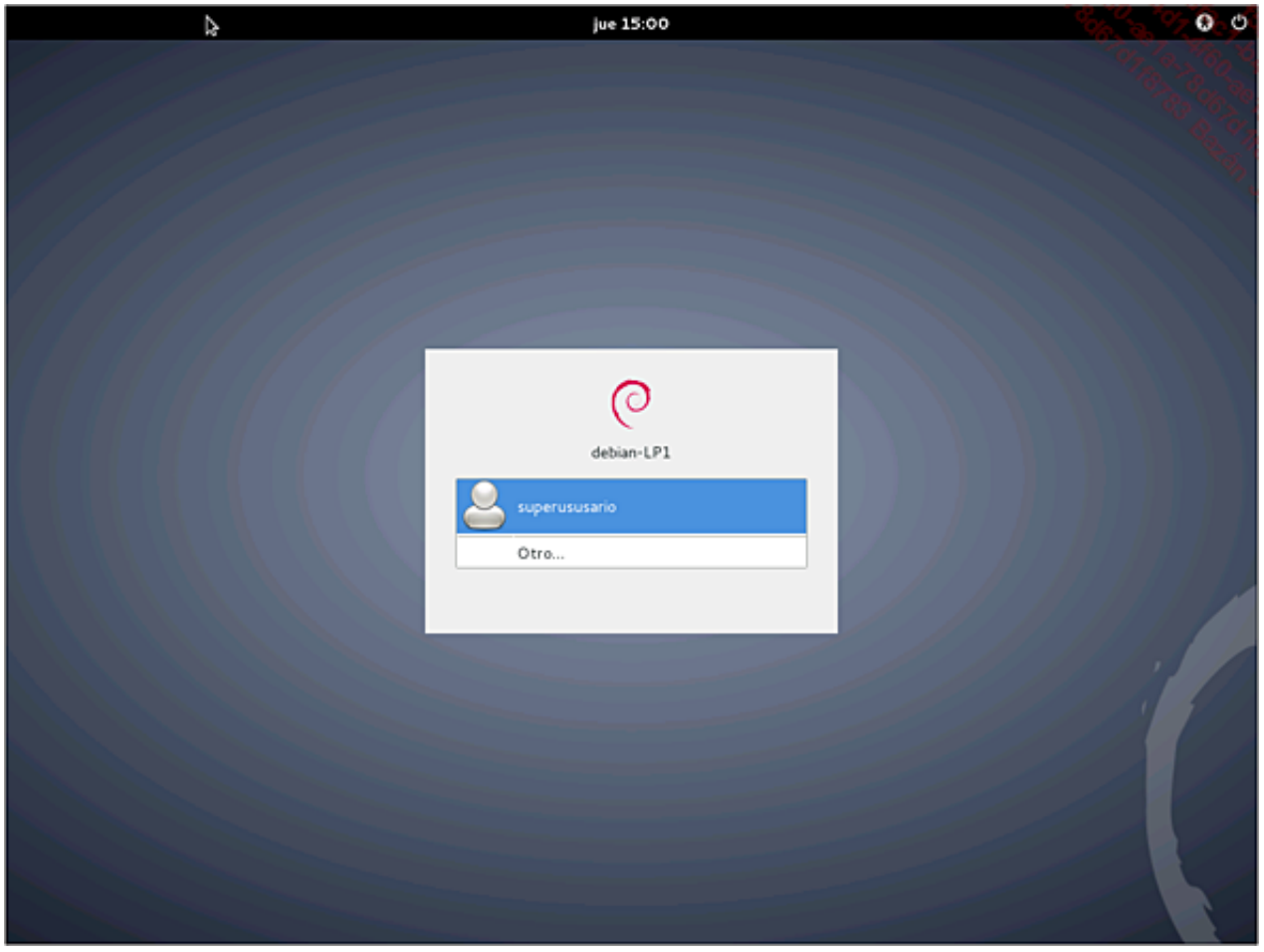
9. Fin de instalación y reinicio

La última etapa antes rematar la instalación es la escritura del gestor de arranque (bootloader). Se trata de la instalación de **GRUB**. Es muy sencillo en este caso, ya que es el único sistema instalado. Pero debe saber que los gestores como Grub invalidarán el instalado anteriormente. Puede utilizar perfectamente grub para arrancar con cualquier sistema, incluido Windows.

Ya sólo falta volver a *bootear*. Grub le propone por defecto dos entradas:

- el modo normal;
- el modo single-user o modo de emergencia.

Inicie en el primero. Si todo va bien y se reconoce la tarjeta gráfica, debe llegar al gestor de sesiones. Cabe destacar que puede modificar el tema visual a su gusto, las elecciones predeterminadas de Debian no son necesariamente las mejores.



Todo marcha correctamente. El gestor de sesiones está a la espera del inicio de sesión.

Instalación de Fedora

1. Soporte de instalación

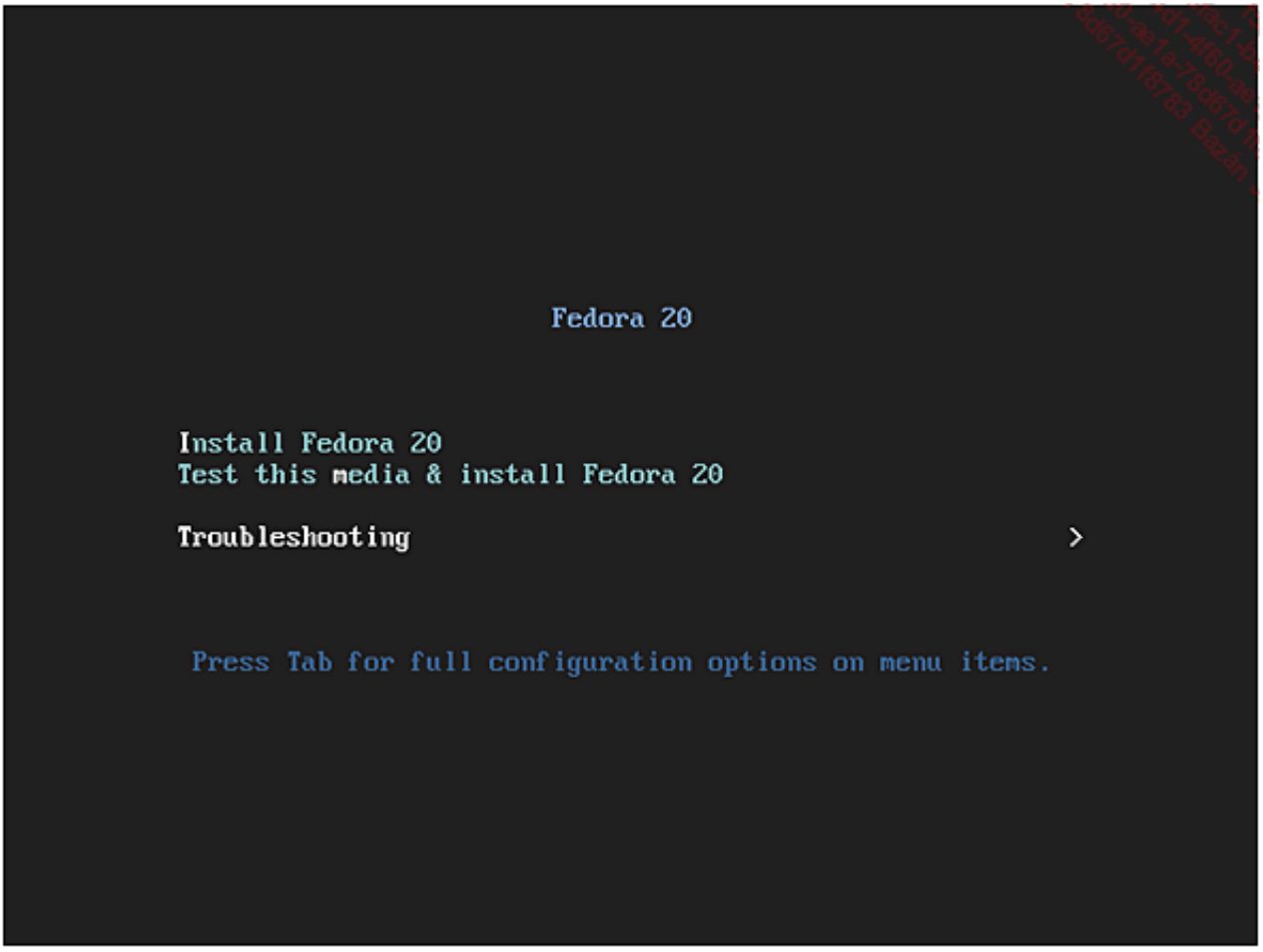
A continuación se muestra el proceso de instalación paso a paso de una distribución basada en paquetes en formato RPM con un instalador en modo gráfico. Siendo menos sobria que la distribución Debian, que es más simple pero más amplia, la instalación de una distribución Fedora se acerca a la de una distribución Red Hat en el ámbito empresarial.

La instalación se ha realizado en una máquina virtual Virtualbox configurada del mismo modo que con la distribución Debian, a excepción de la elección de Fedora 64 bits en el asistente de configuración. Si desea instalar la misma distribución, diríjase al sitio http://www.fedorea.es.org y seleccione **Descargar Fedora - más opciones**.

La instalación se ha realizado desde un DVD de la versión Fedora 20 de 64 bits. También se puede realizar la instalación a través de la red, del mismo modo que con Debian. La imagen ISO puede descargarse desde el siguiente vínculo:http://download.fedoraproject.org/pub/fedora/linux/releases/20/Fedora/x86_64/iso/Fedora-20-x86_64-DVD.iso

2. Arranque del soporte

Configure su ordenador para que arranque a través del soporte e inícielo.



Pantalla de inicio de soporte de la instalación Fedora

La pantalla de inicio del soporte de instalación ofrece una gran cantidad de opciones. La primera entrada permite instalar o actualizar el sistema, la segunda verificar el soporte después de instalar

el sistema. La entrada Troubleshooting permite acceder a un menú suplementario que ofrece una instalación empleando un controlador básico compatible con cualquier tarjeta gráfica, un arranque desde una copia de seguridad del sistema que acompaña al soporte de instalación, o efectuar un test de la memoria. De este modo, se puede reparar su sistema incluso si es imposible iniciarlo desde el disco duro. El arranque con el disco local redirige el arranque a su disco duro.

Una opción curiosa es la comprobación de la memoria. Muchas disfunciones de un equipo se asocian a menudo erróneamente al sistema operativo: pantallas de errores (las famosas BSOD, *Blue Screen Of Death* aunque el color haya cambiado en las últimas versiones) de Windows o kernel panic en Linux. Puede que incluso Linux sea inestable en algunos PC. Ahora bien, con frecuencia el fallo del sistema se debe a un problema de hardware, de los cuales el más común es un problema en uno o varios módulos de memoria. La utilidad memtest+ permite realizar un diagnóstico de dichos módulos mediante pruebas pesadas (y de larga duración). Lo ideal ante un cuelgue inesperado es iniciar este test conservando un solo módulo e ir probando los módulos uno tras otro.

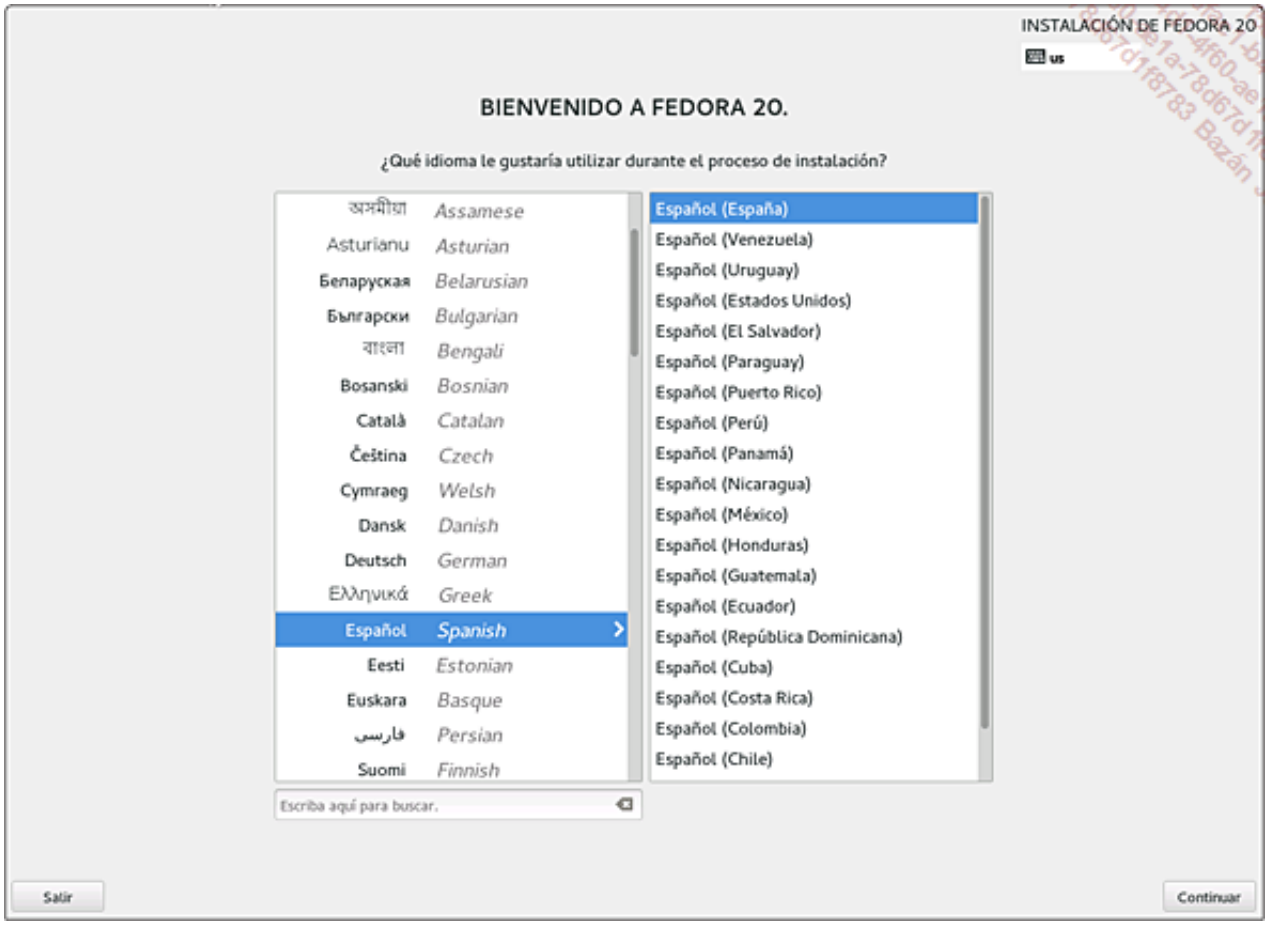
En informática, como en otros ámbitos, la calidad tiene un coste generalmente elevado. Evite componentes de marca "blanca" o genéricos y escoja preferiblemente una marca conocida para todos sus componentes, sobre todo para la memoria (Corsair, Gskill, Kingston, PQI, etc.), las placas base (Gigabyte, Asus, DFI, etc.) y las fuentes de alimentación (Corsair, Cooler Master, Enermax, Fortron, etc.). Asegúrese de la compatibilidad entre sus componentes a través de los vendedores y de la documentación técnica. El autor ha visto módulos de memoria de gran calidad que no funcionan en ciertas placas base y una webcam incluso inactiva bloqueando una tarjeta gráfica cuando se usaba en 3D...

Seleccione la primera opción y pulse [Entrar].

Se iniciará Anaconda, la herramienta de instalación común de Fedora, Red Hat y CentOS.

3. Idioma de la instalación

La herramienta de instalación **Anaconda** puede funcionar en modo texto (consola) o en modo gráfico, que es el que se ha usado en este libro. Tras de diez años sin ninguna modificación, la interfaz de Anaconda ha sido completamente rediseñada para ofrecer un diseño y opciones modernas.

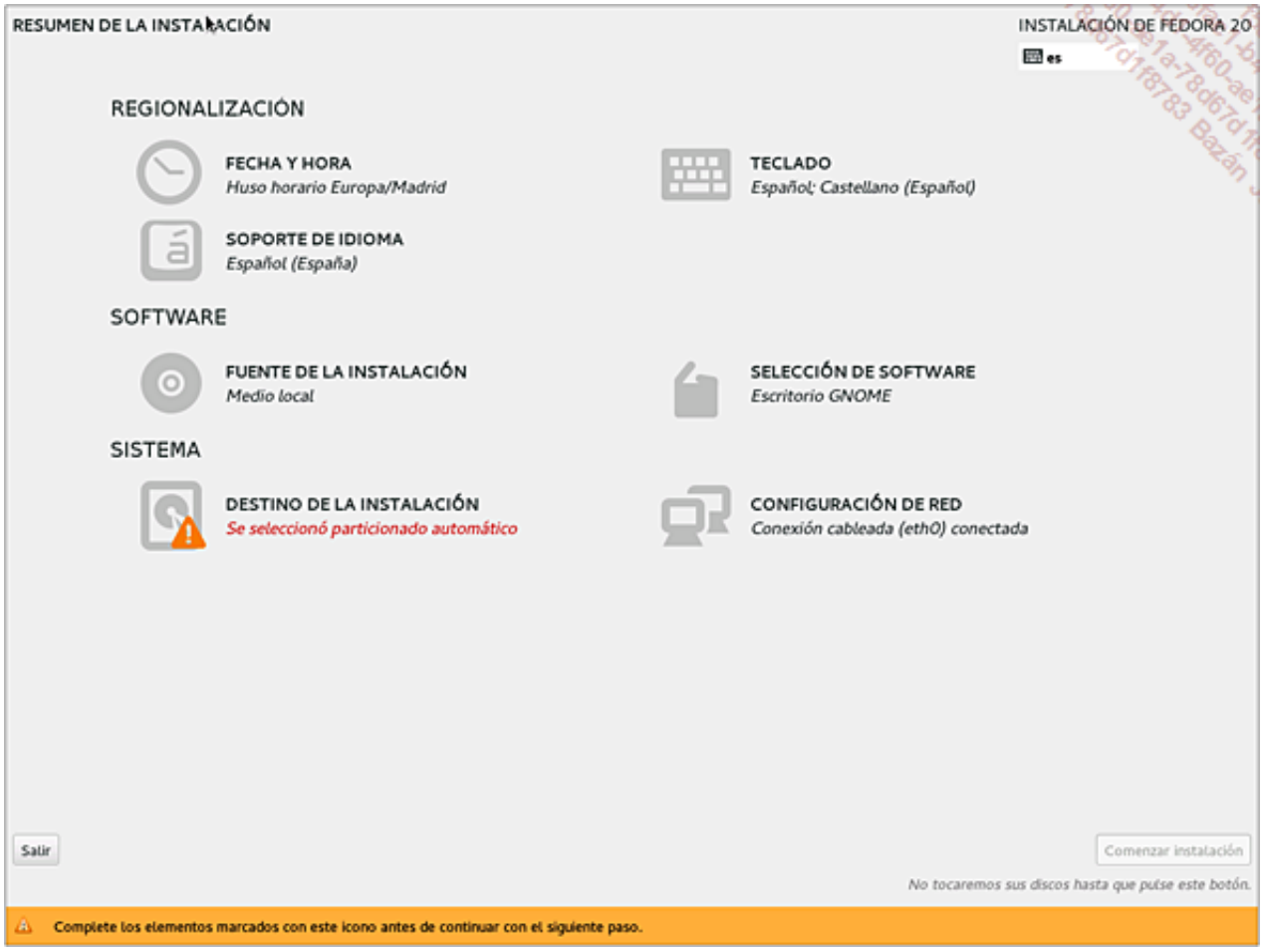


Idioma por defecto de Fedora

Debe seleccionar el idioma de instalación, en este caso **Español** (**Spanish**) a la derecha en **Español**(**España**) y hacer clic en **Continuar**. A partir de este momento, la interfaz se traducirá al castellano y se mantendrá en el idioma seleccionado hasta que finalice el proceso de instalación.

4. Resumen de la instalación

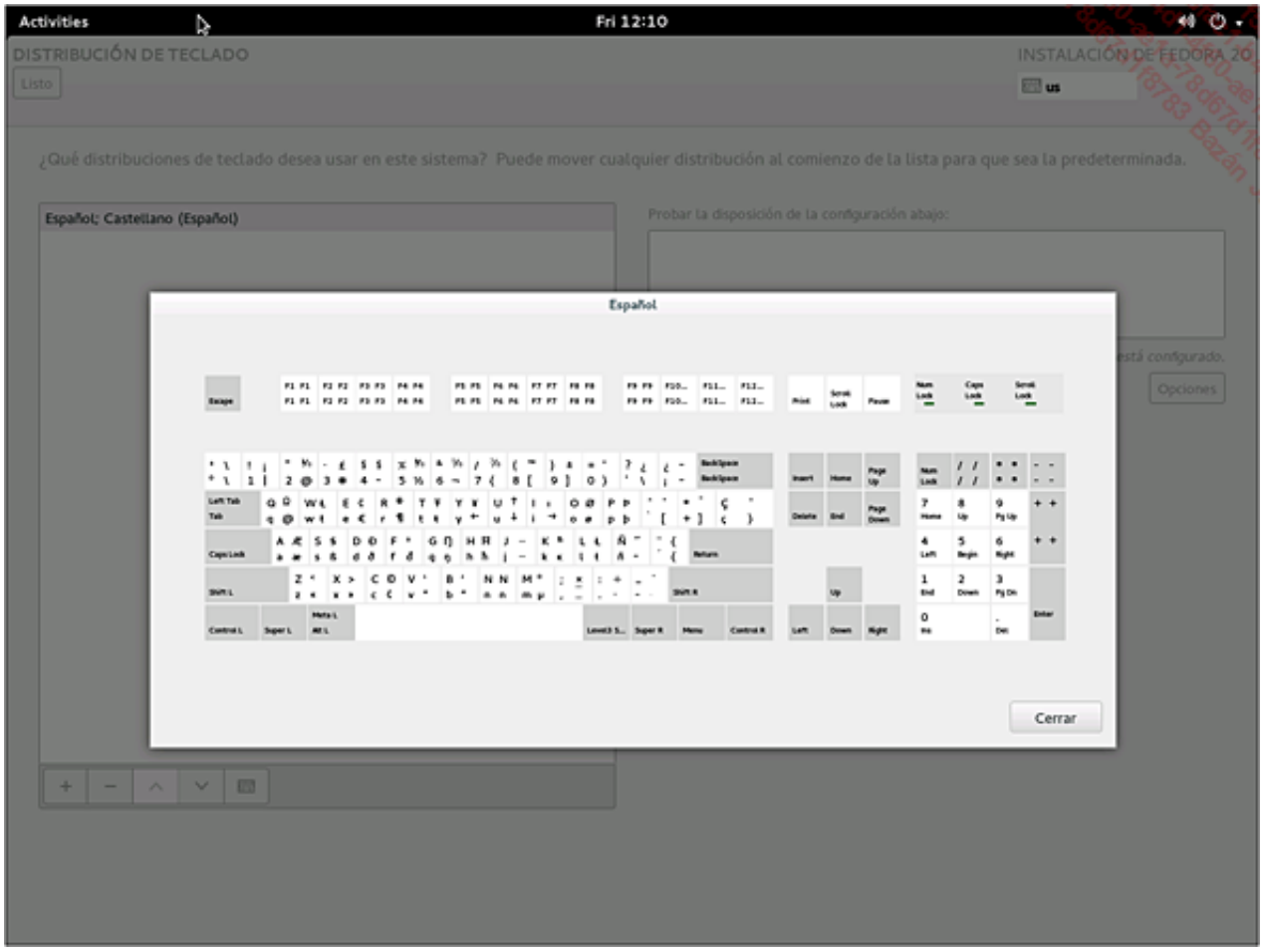
La pantalla siguiente ofrece un resumen de la instalación. Es la pantalla principal en la que, empleando los iconos, escogerá las diferentes opciones de instalación. El punto de exclamación que aparece en ciertos iconos indica las opciones obligatorias: deberá completar los puntos asociados antes de pasar a la etapa siguiente. Generalmente un solo punto es importante, como en Debian: el particionado. Aquí las etapas detalladas.



Resumen de la instalación

5. Teclado

Haga clic en **Teclado**. El método escogido por Fedora es muy inteligente. Podemos efectuar una selección manual, al igual que en todos los instaladores. A la izquierda, haciendo clic en el signo [+], puede seleccionar su propia distribución de teclado. Esto es muy inteligente, le permite mostrar la disposición del teclado haciendo clic en el icono correspondiente: imposible de equivocarse, basta con comparar con su teclado físico. El autor por ejemplo dispone de un teclado xxxxxxx. En este caso, la opción es **Español (xxxxxxx)**. Par un teclado PC común, la opción **Español (variante)** o**Español (variante latin0)** es ideal.



La opción del teclado es una de las mejores en Linux

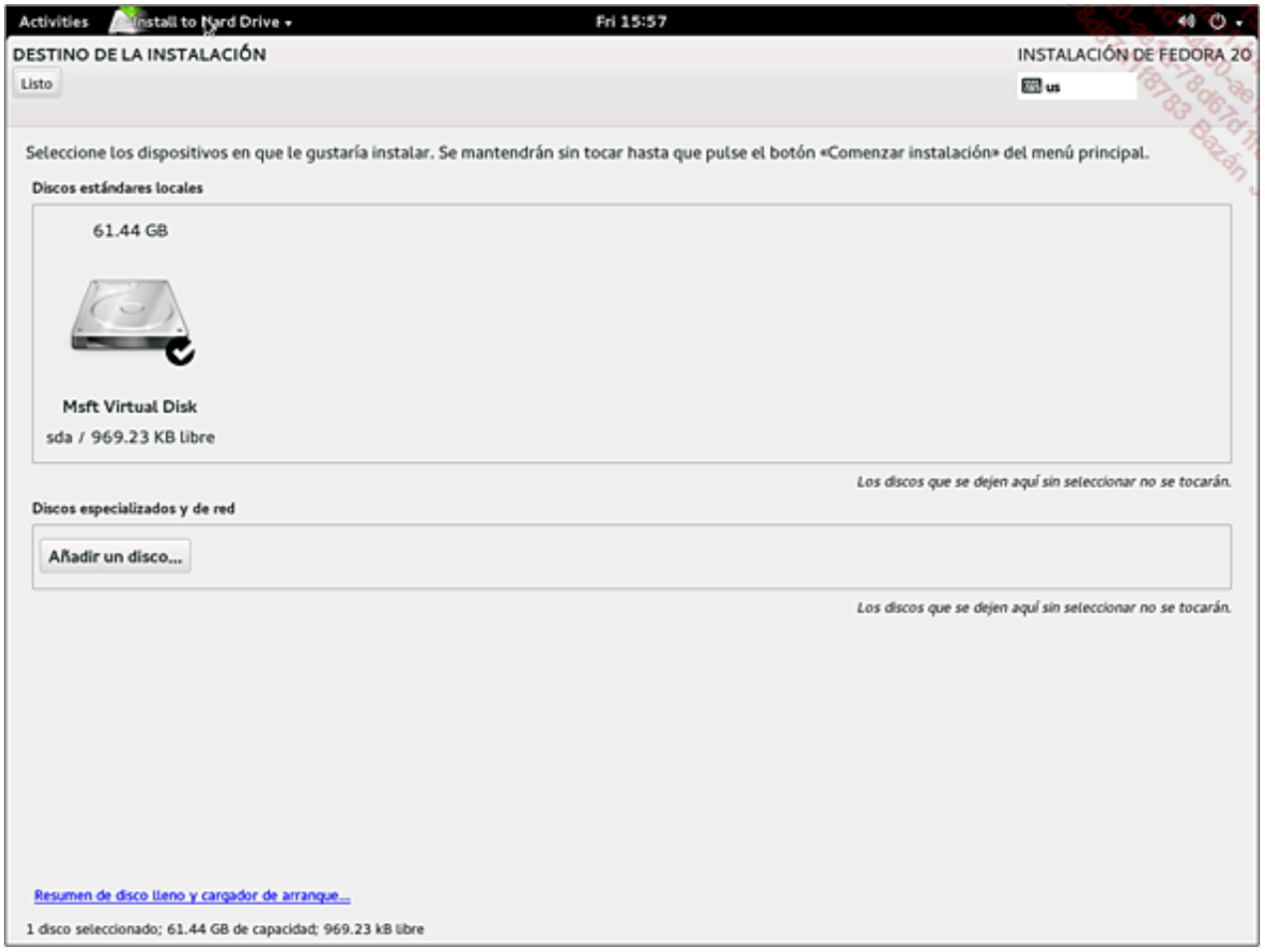
Una vez efectuada la selección de los teclados, puede modificar el orden predeterminado. El campo de introducción de la derecha permite comprobar la opción del teclado.

Una última cosa: arriba a la derecha de las pantallas de instalación encontramos el teclado actual. Para cambiar de un teclado a otro (si ha seleccionado varios), basta con hacer clic aquí.

Luego haga clic en **Terminar**.

6. Destino de instalación

Haga clic en **Destino de la instalación**. Esto efectuará la elección del particionado. La pantalla ofrece dos marcos: los discos locales arriba y los discos especiales abajo. Esta opción puede sorprender a un neófito pero simplemente recuerda que Fedora sirve de base para las distribuciones Red Hat profesionales y ofrece elecciones relativas a este contexto.

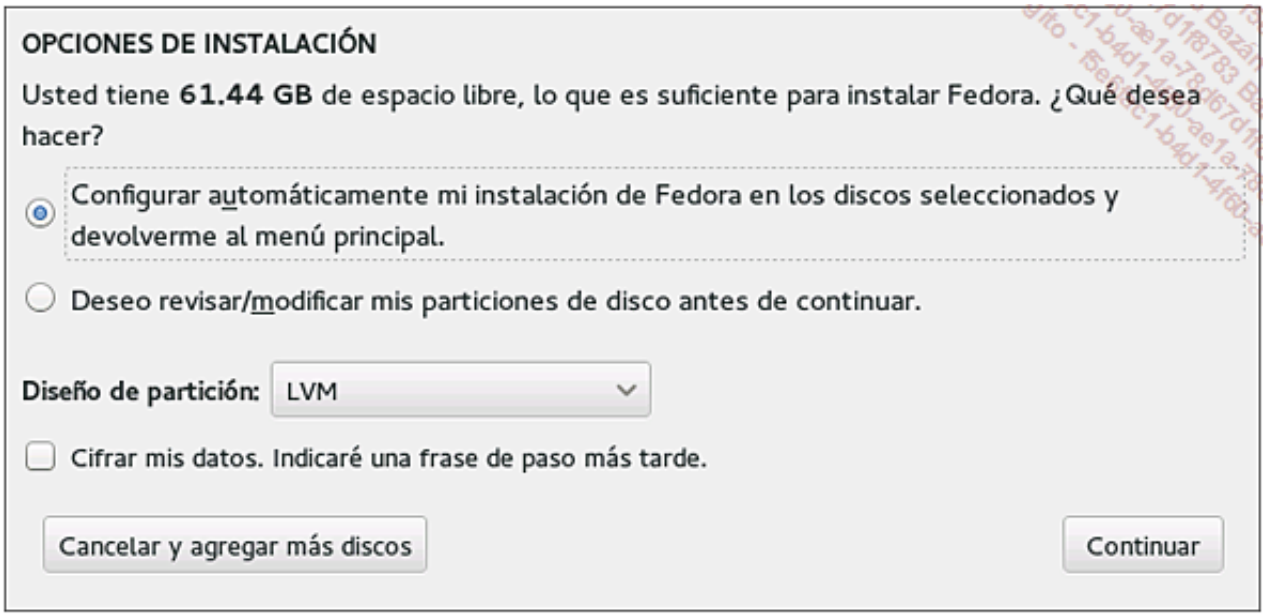


Selección del soporte de destino

En una máquina virtual o en su PC de escritorio, es evidente determinar a qué va dirigida la primera opción: la instalación se realizará en uno de los discos conectados a su PC, fundamentalmente un disco SATA local, SCSI o IDE para los equipos más antiguos. La segunda opción es útil en ámbitos empresariales donde el sistema se puede instalar en un disco SAN, RAID por hardware, Multipath o iSCSI.

Fedora tiene en cuenta los soportes de almacenamiento empresariales

Compruebe que su disco duro está seleccionado y haga clic en **Listo**. Se muestra un cuadro de diálogo:



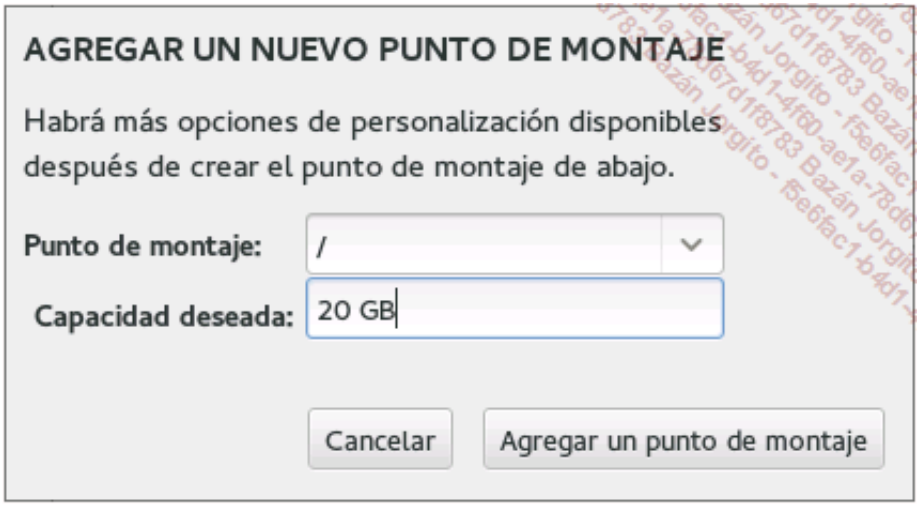
Selección de esquema de particionado

Vamos a efectuar un particionado manual. Para ello, seleccione **Deseo revisar/modificar...** y el diseño de partición **Partition Standard**. Luego haga clic en **Continuar**.

La pantalla de particionado manual es tan inteligente como la del teclado. Puede solicitar a Anaconda crear un diseño haciendo clic en el vínculo ofrecido. Sin embargo, crearemos el diseño siguiente, partiendo de un volumen de unos 60 GB:

- Raiz de 20 GB
- /home de 38 GB
- Swap de 2 GB

Para ello, haga clic en el signo +, utilice / como punto de montaje, indique 20 GB como tamaño y luego haga clic en **Agregar un punto de montaje**.

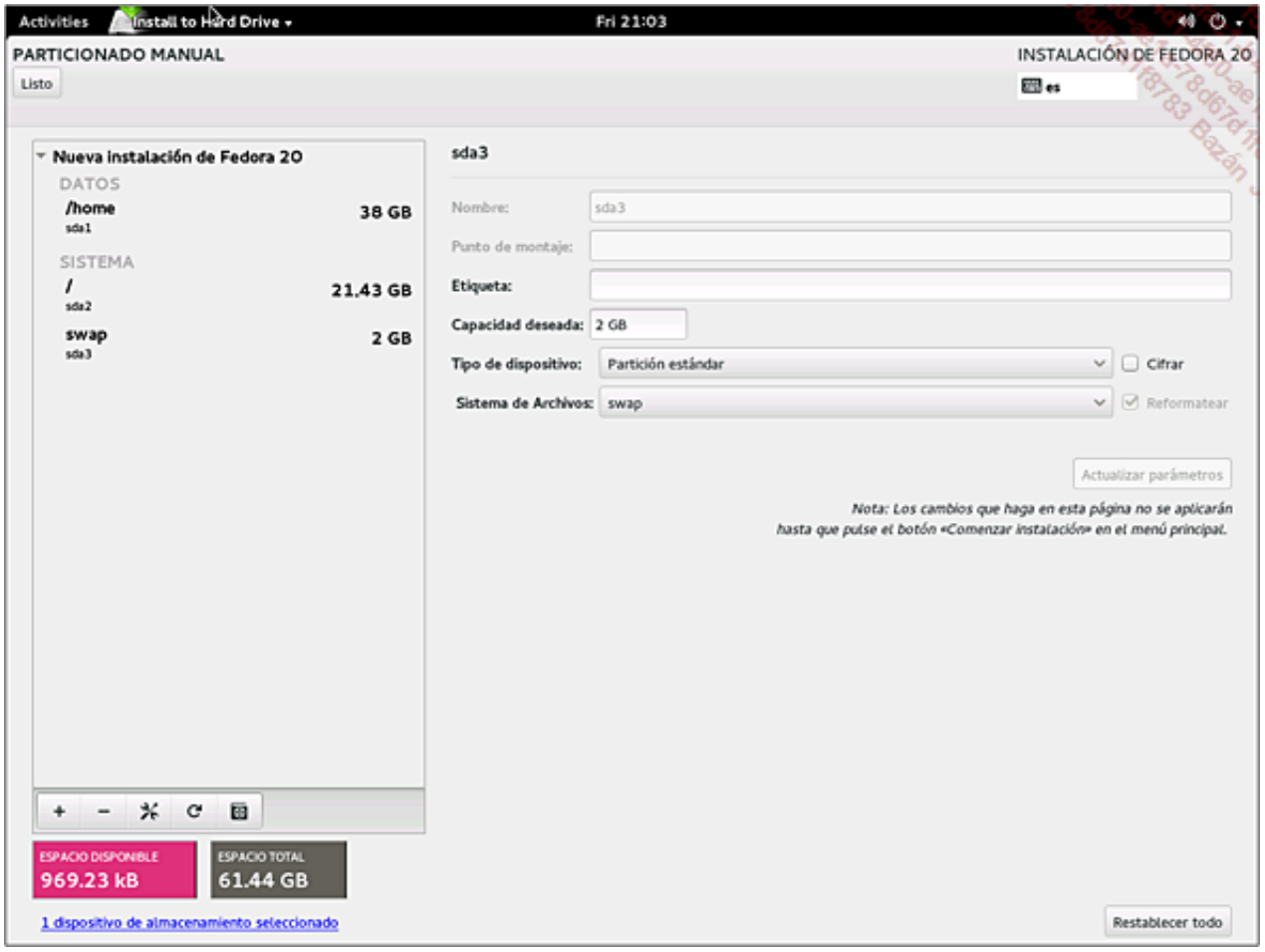


Creación de la partición raíz

De vuelta a la pantalla de particionado manual, verifique los valores situados a la derecha cuando seleccione su partición. Puede modificar el punto de montaje, el tamaño, o agregar una etiqueta. Una opción importante es el tipo de sistema de archivos. Mantenga el tipo **ext4**.

Repita la operación para /home y swap, sabiendo que para ésta última el punto de montaje es**swap**.

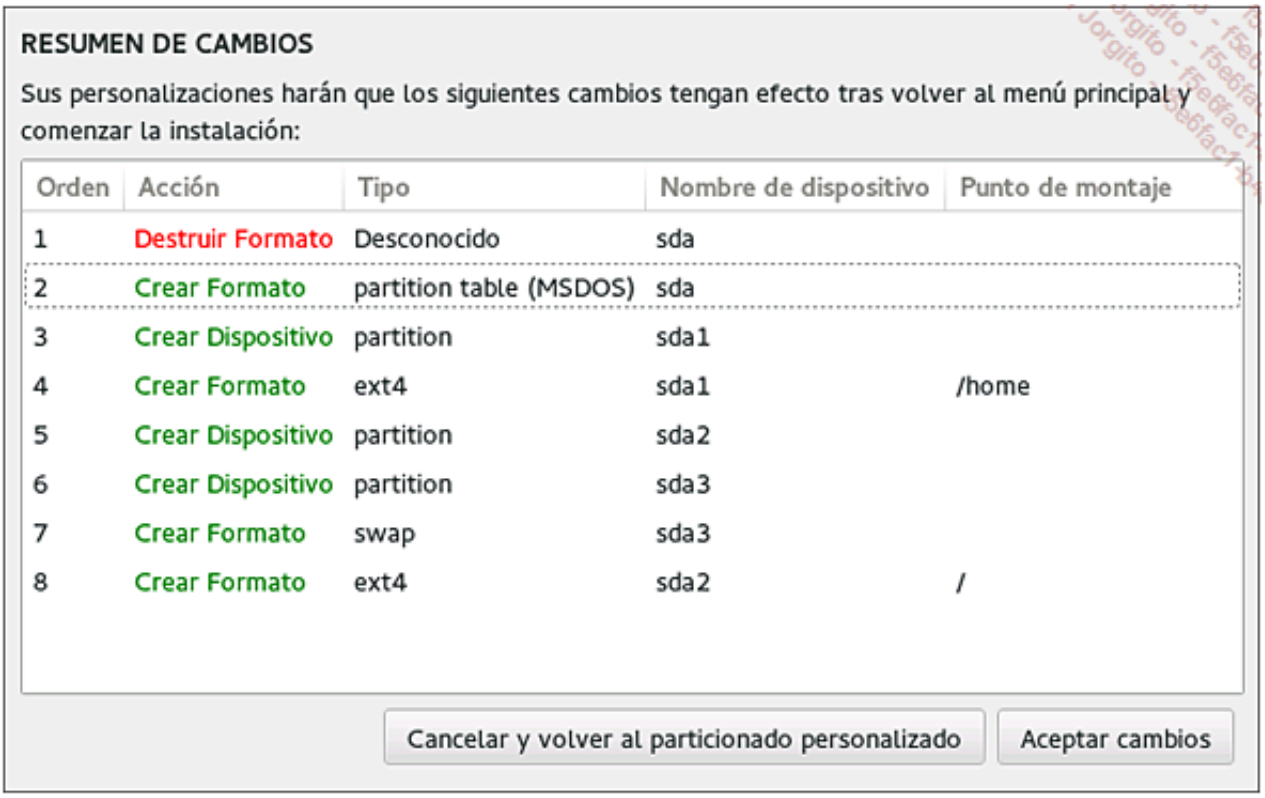
Es posible que tenga que efectuar algunos ajustes. Las capacidades utilizadas y disponibles figuran en la parte inferior de la pantalla. Aquí vemos el diseño final:



Diseño de particionado personalizado

Cuidado: haga clic en **Listo** arriba a la izquierda para seleccionar su diseño, y no en **Restablecer todo** abajo a la derecha.

A continuación haga clic en **Aceptar cambios**.



Diseño de particionado definitivo.

La siguiente etapa permite elegir el nombre de anfitrión y, si se considera oportuno, con el botón**Configurar la Red**, modificar los parámetros asociados. Es aquí donde podrá configurar sus interfaces de red de forma estática o por DHCP.

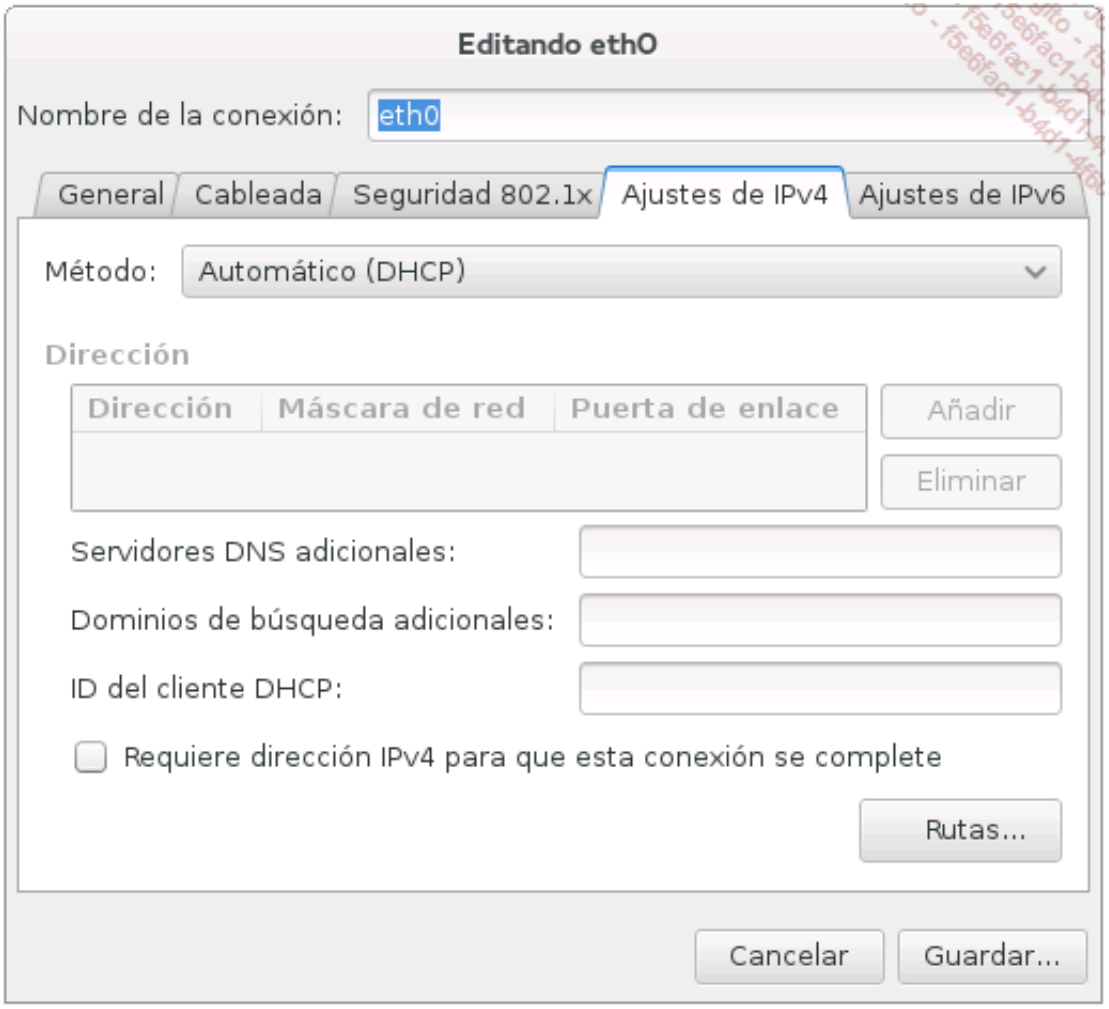
7. Configuración de red

Esta etapa es opcional para una estación de trabajo o un equipo doméstico.

Haga clic en el botón **Configuración de Red**. Aparecen los adaptadores detectados, Ethernet o Wi-Fi. Modifique si lo desea, abajo a la izquierda el FQDN (nombre de su host y nombre de dominio) de su equipo. El botón **+** permite agregar un dispositivo (una noción avanzada que permite agregar varios adaptadores en uno solo) o una VLAN (será necesario conocer el identificador de la VLAN deseada).

Con un clic en el adaptador y luego en **Configurar** abajo a la derecha, puede modificar el comportamiento del adaptador, como su dirección IP estática o DHCP.

En el cuadro de diálogo **Conexiones de red**, compruebe que está en la pestaña **Cableada**, haga clic en su interfaz y a continuación pulse **Editar**. Vaya a la pestaña deseada, por ejemplo **Ajustes de IPv4**, y modifique sus datos IP (puerta de enlace estática, introducción de sus direcciones IP, etc.).

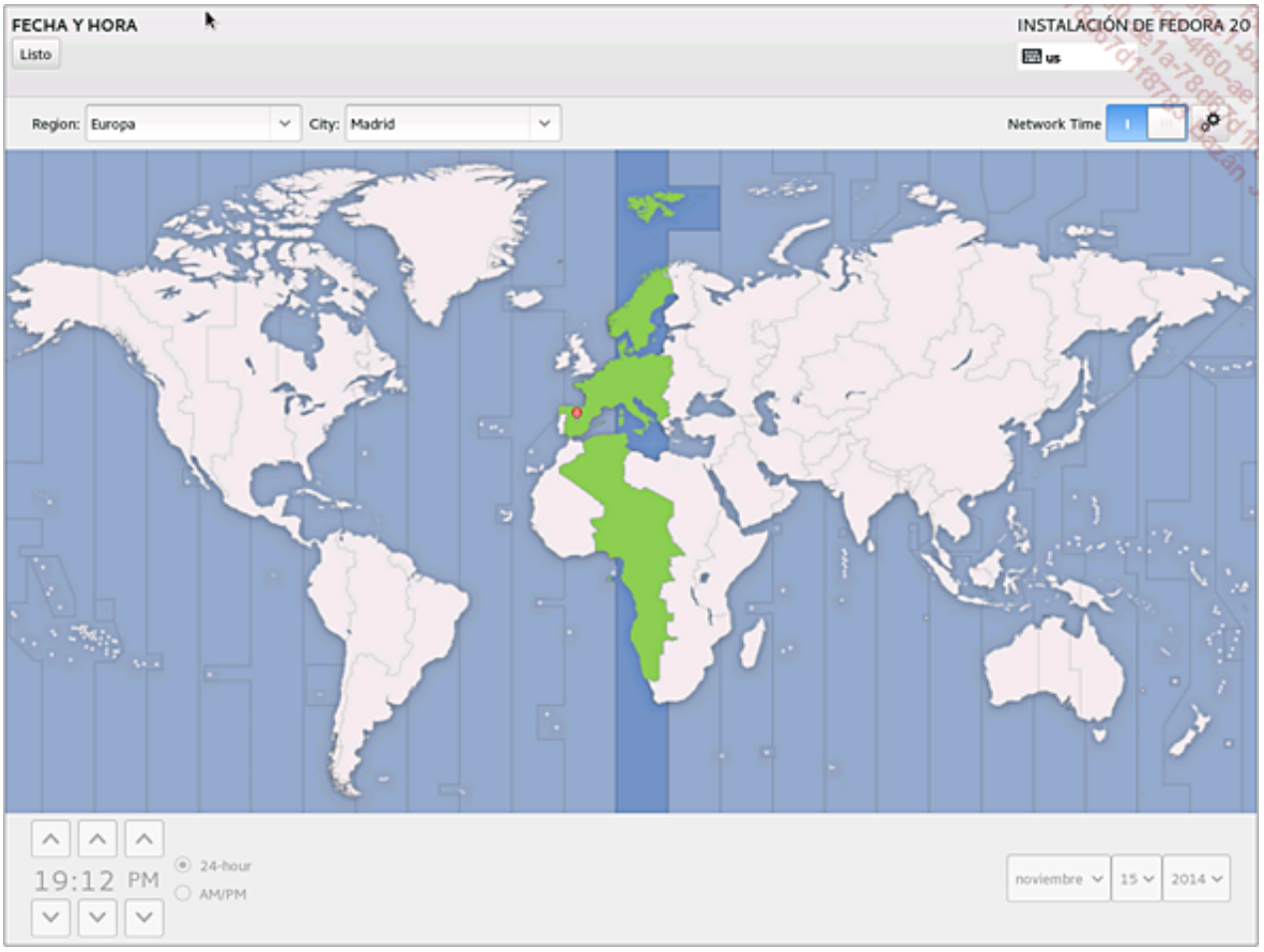


Configuración avanzada de una interfaz de red

Haga clic en **Listo** una vez haya finalizado el procedimiento de configuración.

8. Reloj

Haga clic en **Fecha y hora**. Seleccione el huso horario correspondiente a su país o ciudad con un clic o simplemente seleccionando su continente y ciudad. Si cuenta con una conexión de red, observará que NTP (protocolo que permite una actualización de la hora automática de su equipo) está activada por defecto.

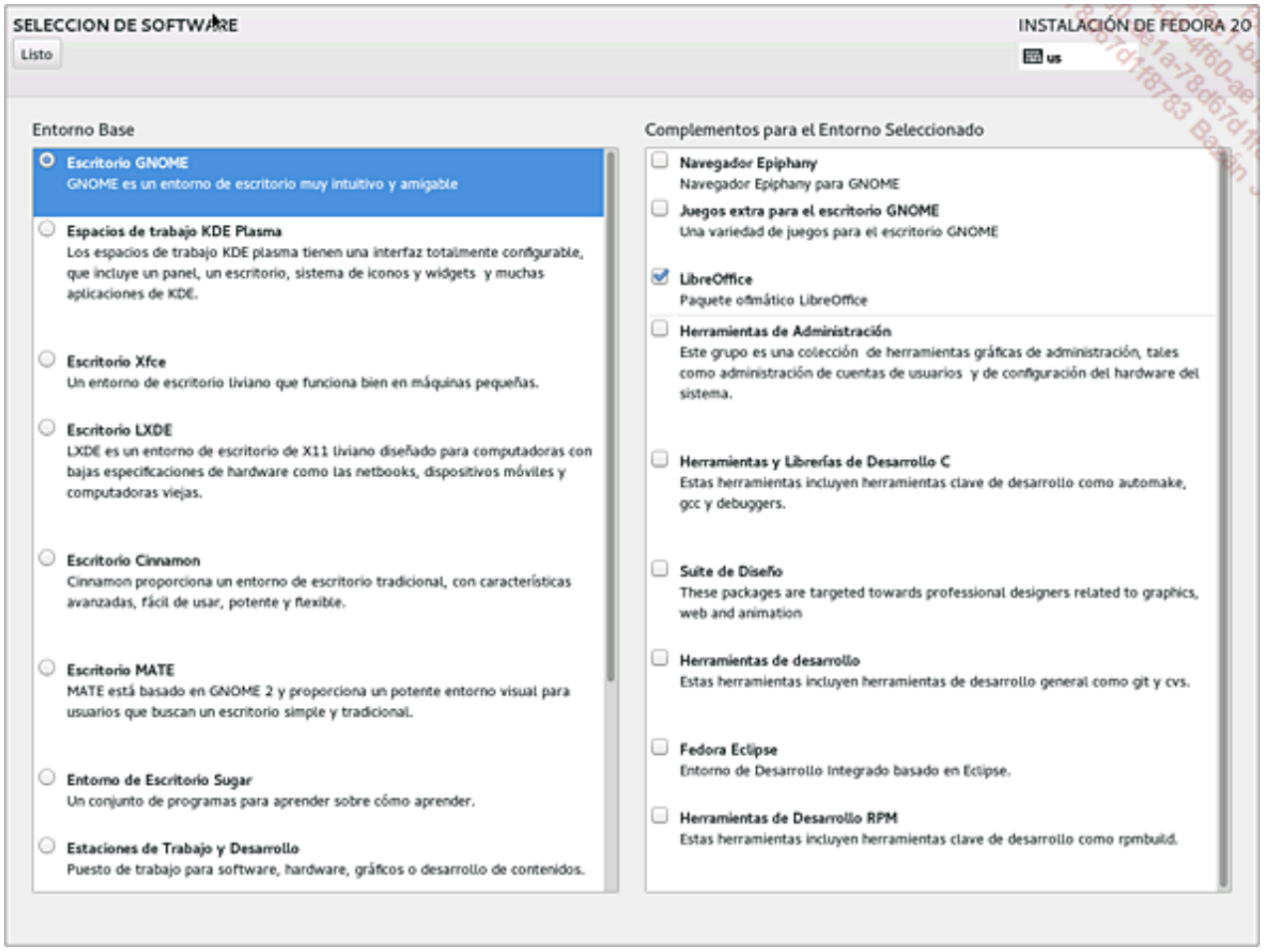


Configuración de la fecha y hora

Haga clic en **Listo**.

9. Selección de paquetes

Haga clic en **Selección de Software**. La selección de software es muy lógica. La elección de un entorno base determina los productos que serán instalados de forma predeterminada. Seleccione**Escritorio GNOME**. Una opción sensata para un servidor será seleccionar una instalación mínima y luego personalizar el sistema una vez este se instale.



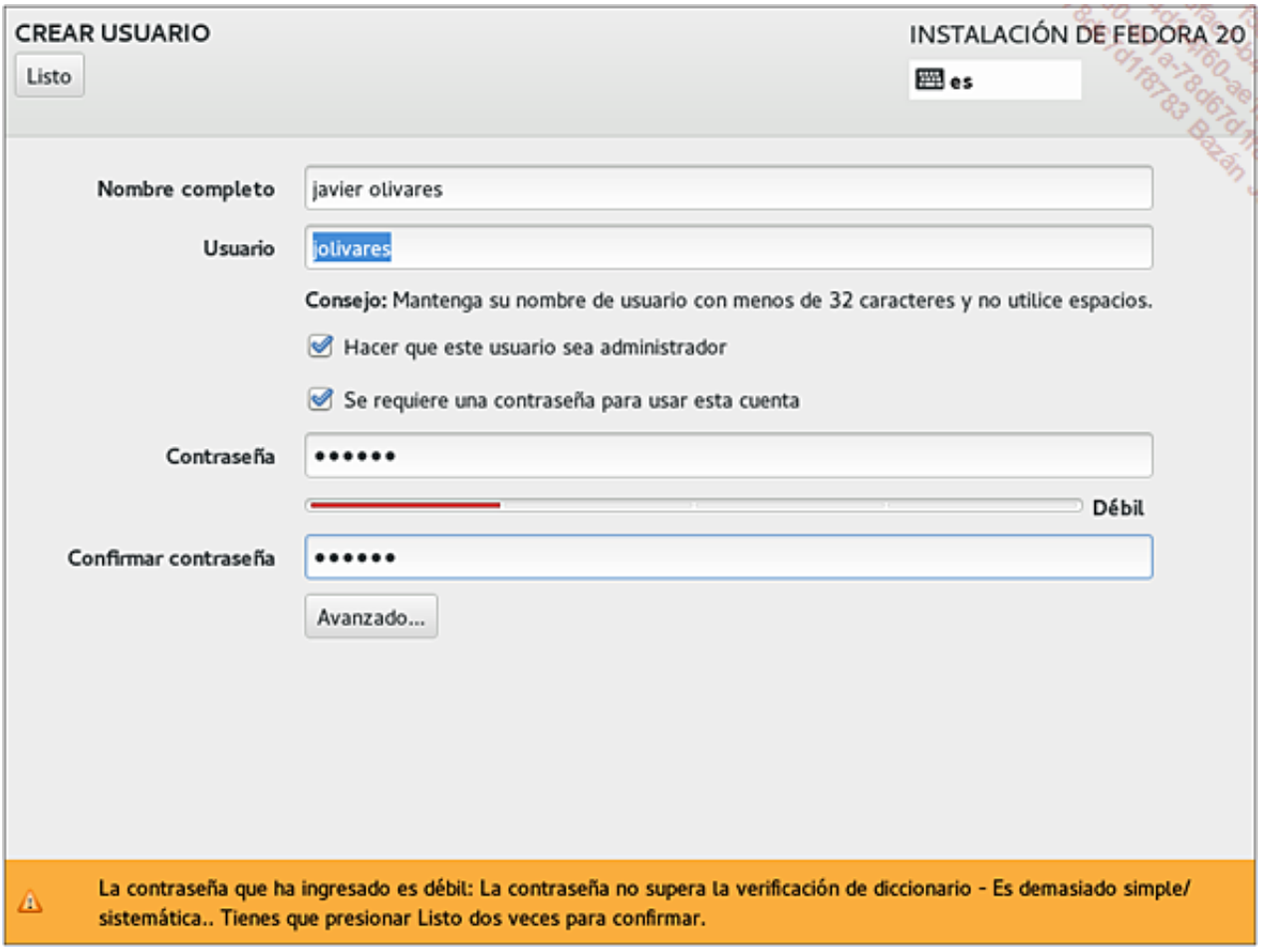
Software a instalar

Haga clic en **Listo** y luego en **Comenzar instalación**.

10. Usuarios

Mientras la instalación se lleva a cabo, deberá escoger una contraseña para el administrador (contraseña de root) y crear un usuario. Si la contraseña root es demasiado simple, deberá hacer clic dos veces en **Listo** para validarla.

La creación de un usuario no es obligatoria pero si aconsejable. El procedimiento es simple. Puede crear un administrador, y al igual que para root deberá validar dos veces si la contraseña es muy simple.



Agregar un usuario administrador

11. Fin de la instalación

Al terminar la instalación básica, haga clic en **Terminar la instalación** abajo a la derecha. Las etapas post-instalación se desarrollan: escritura de los archivos de configuración, adaptación de initramfs en función de su configuración, implementación del gestor de arranque (boot). Solo faltará hacer clic en **Reiniciar**.

Es sistema reinicia y se despliega muy sobrio el administrador de visualización.

El administrador de visualización en el primer arranque

Red Hat Package Manager

1. Noción de paquete

Al contrario de otros sistemas operativos, con Linux y Unix no es habitual disponer de software proporcionado con un programa de instalación interactivo (y no install.exe). Algunos editores proponen scripts de instalación y, muy a menudo, éstos sólo descomprimen y desarchivan algunos archivos.

Con Linux es muy habitual disponer de varios productos, herramientas, actualizaciones, etc. en forma de paquetes (packages). Un paquete es un archivo que contiene el producto a instalar y unas reglas. Según su contenido, su tamaño puede ser muy imponente. El núcleo y todos sus módulos son, por ejemplo proporcionados en esta forma. Las reglas pueden ser múltiples:

- Gestión de las dependencias: sólo se podrá instalar el producto si los productos que él mismo utiliza están ya presentes.
- Preinstalación: se deben prever acciones antes de poder instalar el producto (cambiar derechos, crear directorios, etc.).
- Postinstalación: se deben prever acciones después de la instalación del producto (parámetros de un archivo de configuración, compilación anexa, etc.).

En Red Hat, Fedora, SuSE, Mandriva y algunas otras distribuciones el formato de paquete por defecto es el **RPM** (*Red Hat Package Manager*). En Debian, Ubuntu, es el formato **DPKG** (*Debian Package*). Además de por el formato, se diferencian principalmente por las herramientas.

El hecho de disponer de la información de dependencias permite obtener herramientas eficaces que pueden solas resolverlas en cascada. Al instalar un paquete, la herramienta podrá instalar todas las dependencias necesarias. A veces se pueden especificar varias ubicaciones (repositorios) para estos paquetes, ya sean locales (disco duro, CD-Rom, DVD, etc.) o remotos (http, ftp, etc.).

Debemos favorecer el uso de un paquete previsto para su distribución cuando esté disponible. Si no es el caso, a veces se puede utilizar un paquete de un producto competidor o volver a compilar el producto por sí mismo.

Se simplifican mucho las actualizaciones de un sistema Linux que utilizan un sistema de empaquetado. Para pasar de la versión de un producto a otra, basta con recuperar el paquete del producto en versión superior e instalarlo. Todas las actualizaciones siguen este principio. Existe un formato, de **delta-rpm**, que sólo proporciona en el paquete las diferencias de una versión a otra. Pero siempre es posible utilizar un paquete completo.

2. El gestor RPM

RPM es un gestor de paquetes desarrollado por Red Hat que luego ha sido adoptado de manera masiva por muchas otras distribuciones. Simplifica mucho la distribución, la instalación, la actualización y la supresión de los programas. Se basa en comandos (p. ej.: **rpm**), una base de datos local y paquetes en formato rpm (extensión rpm).

La base de datos está ubicada en **`/var/lib/rpm`**. Contiene toda la información correspondiente a los programas instalados, sus versiones, sus archivos y derechos, y sus dependencias. A no ser que se produzca un grave problema, no se debe modificar NUNCA esta base de datos. Hay que utilizar las herramientas RPM.

Se facilita cada programa en forma de paquete con formato RPM. El rpm responde a una nomenclatura precisa.

```
nombre-versión-edición.arquitectura.rpm
```

por ejemplo:

```
php-4.1.2-2.1.8.i586.rpm
```

La edición es un identificador de versión del paquete RPM propio del editor. Aquí tenemos la versión 2.1.8 del paquete PHP versión 4.1.2. La arquitectura es i586 (Intel Pentium). También se puede encontrar i386, i686, x86_64 o amd64 (64 bits), ppc64, s390x, arm o arm32, arm64 o noarch. Un paquete noarch no contiene programas o librerías binarias, sino código independiente como scripts, documentación, imágenes, sonido, vídeo, etc.

3. Instalación, actualización y eliminación

Se instala un paquete rpm con el parámetro `-i`.

```
rpm -i php-4.1.2-2.1.8.i586.rpm
```

Dado que es posible utilizar comodines (**`rpm -i *.rpm`**), quizás necesite visualizar el nombre del paquete durante la instalación; use el parámetro `-v`. El parámetro `-h` visualiza caracteres **#** para indicar la progresión de la instalación. La instalación no funcionará si no se resuelven las dependencias.

La actualización de un producto hacia una versión superior desde un paquete se hace con el parámetro `-U`. En este caso, todos los archivos se actualizan con los de la nueva versión: se suprimen los antiguos y los nuevos los sustituyen. Se guardan los antiguos archivos de configuración con la extensión **`.rpmsave`**. Si el paquete no está instalado, la actualización asume el papel de instalación. Cuidado con este parámetro: instala el paquete incluso si la versión que actualiza no está instalada.

```
rpm -Uvh php-4.1.3-1.i586.rpm
```

La actualización es también posible con `-F`. Pero si el paquete no está instalado, tampoco se instalará durante la actualización, al contrario que con `-U`. Así, si dispone de todos los paquetes de actualización del sistema y usted sólo desea actualizar los que realmente están instalados, entonces puede teclear:

```
rpm -Fvh *.rpm
```

La supresión se efectúa con el parámetro `-e`. Sin embargo, observe que deberá pasar como argumento el nombre del paquete, y no el nombre del archivo de paquete.

```
rpm -e php
```

Cuenta con varias opciones:

- **`--force`**: en caso de conflicto con otro paquete (el caso más habitual es el de dos paquetes que proponen el mismo archivo en el mismo sitio), esta opción fuerza la instalación.
- **`--nodeps`**: si el paquete rechaza instalarse por culpa de un problema de dependencias, esta opción forzará la instalación. Ocurre a veces que este error se produce cuando se instala la dependencia en cuestión de otra manera y no es a partir de un paquete rpm (p. ej.: compilación, binario copiado a mano).

En la medida de lo posible, evite utilizar estas opciones, que pueden romper algunas dependencias, en particular si utiliza un sistema de metapaquete (yum, apt, zypper, urpmi, etc.).

4. El caso del núcleo

La instalación o la actualización de un núcleo es un caso particular. En efecto, la actualización suprime la versión antigua. El núcleo es un componente muy crítico del sistema. Si, como resultado de la actualización, el sistema no funciona (o funciona mal), habría que volver a instalar un antiguo núcleo desde el soporte de instalación. El procedimiento es el siguiente:

- Instalación del nuevo núcleo con el parámetro `-i`, se añadirá al sistema.
- Reinicio y prueba de los programas y periféricos con el nuevo núcleo.
- Si funciona correctamente, posible eliminación del antiguo núcleo con `-e`.
- Edición de **/boot/grub/grub.conf** y modificación de la línea **Default** para arrancar por defecto en el nuevo núcleo si utiliza la primera versión de grub, o actualiza el cargador de arranque empleando sus herramientas.

5. Consultas RPM

Se puede consultar la base de datos RPM con el parámetro `-q` seguido de algunas opciones.

-a: lista de todos los paquetes instalados.

-i: información general (el resumen) del paquete.

-l: lista de los archivos instalados.

-f nombre: encuentra el paquete que contiene el archivo dado.

-p nombre: la búsqueda se efectúa en el archivo del paquete dado.

--requires: dependencias del paquete.

--provides: lo que proporciona el paquete.

--scripts: scripts ejecutados en la instalación y la supresión.

--changelog: el historial del paquete.

```
$ rpm -qilp libjpeg-6.2.0-738.i586.rpm
Name        : libjpeg                      Relocations: (not relocateable)
Version     : 6.2.0                             Vendor: SUSE LINUX Products
GmbH, Nuernberg, Germany
Release     : 738                           Build Date: Sat Mar 19 20:07:55
2005
Install date: (not installed)               Build Host: d121.suse.de
Group       : System/Libraries              Source RPM: jpeg-6b-738.src.rpm
Size        : 125804                           License: BSD, Other License(s),
see package
Signature   : DSA/SHA1, Sat Mar 19 20:12:06 2005, Key ID a84edae89c800aca
Packager    : http://www.suse.de/feedback
URL         : http://www.ijg.org/
Summary     : JPEG libraries
Description :
The libraries (static and dynamic) for the jpeg-graphics format. The
sources are contained in the jpeg source package.



Authors:
--------
    Rob Hooft <hooft@EMBL-Heidelberg.DE>
    Michael Mauldin <mlm@cs.cmu.edu>
/usr/lib/libjpeg.so.62
/usr/lib/libjpeg.so.62.0.0
```

```
$ rpm -qp libjpeg-6.2.0-738.i586.rpm -requires
/sbin/ldconfig
/sbin/ldconfig
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(CompressedFileNames) <= 3.0.4-1
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1.3)
rpmlib(PayloadIsBzip2) <= 3.0.5-1

$ rpm -qi php
Name        : php                           Relocations: (not relocateable)
Version     : 4.1.2                              Vendor: Red Hat, Inc.
Release     : 2.1.8                          Build Date: mer 14 jui 2004
11:24:16 CEST
Install date: lun 27 jun 2005 19:36:32 CEST      Build Host: porky.build.
redhat.com
Group       : Development/Languages         Source RPM: php-4.1.2-2.1.8.src.
rpm
Size        : 3843552                          License: The PHP License,
version 2.02
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://www.php.net/
Summary     : The PHP HTML-embedded scripting language. (PHP: Hypertext
Preprocessor)
Description :
PHP is an HTML-embedded scripting language. PHP attempts to make it
easy for developers to write dynamically generated webpages. PHP also
offers built-in database integration for several commercial and
non-commercial database management systems, so writing a
database-enabled webpage with PHP is fairly simple. The most common
use of PHP coding is probably as a replacement for CGI scripts. The
mod_php module enables the Apache Web server to understand and process
the embedded PHP language in Web pages.
$ rpm -qa | grep php
php-ldap-4.1.2-2.1.8
php-imap-4.1.2-2.1.8
asp2php-0.75.17-1
php-4.1.2-2.1.8
$ rpm -qf /usr/bin/passwd
passwd-0.68-1.2.1
```

6. Verificación de los paquetes

Es posible que, después de la instalación de un paquete, se haya alterado uno o varios archivos instalados (cambio de permisos, de propietario, edición, supresión, etc.). Como la base RPM contiene toda la información necesaria, se puede pedir una comprobación con el parámetro `-V`.

```
$ rpm -V php
S.5....T c /etc/php.ini
```

Un punto significa que una etapa de comprobación está OK. Si no es así:

- **S**: se modificó el tamaño del archivo.
- **5**: la suma MD5 ya no se corresponde.
- **T**: la fecha de modificación ya no es la misma.
- **U**: se modificó el propietario.
- **G**: se modificó el grupo.

- **L**: se modificó el vínculo simbólico.
- **M**: se modificaron los permisos o el tipo de archivo.
- **D**: se modificó el periférico (major/minor).

La **c** indica que se trata de un archivo de configuración.

A menudo el editor de la distribución firma los archivos de paquetes RPM para garantizar su integridad. Se puede comprobar la integridad de un paquete con una clave pública GPG que se tiene que haber cargado ya en el sistema.

```
gpg --import RPM-GPG-KEY
rpm --import RPM-GPG-KEY
rpm --checksig libjpeg-6.2.0-738.i586.rpm
```

7. Las dependencias

Si usted utiliza las herramientas gráficas que proporciona su distribución, éstas intentarán resolver las dependencias en lugar de usted. El comando **rpm** solo no lo hace por defecto. Hay herramientas complementarias "frontend" como **yast**, **apt** o **yum** que lo hacen en su lugar. La distribución Red Hat facilitaba hasta las versiones 4 (RHEL) una herramienta llamada **rpmdb-redhat** para instalar de manera automática las dependencias mediante **rpm** (el empleo de yum a partir de RHEL 5 ha dejado esta herramienta obsoleta). Esto implica en particular el hecho de que todos los paquetes de la distribución deben estar en el mismo lugar (en el mismo directorio) y que el sistema sólo funciona con los paquetes oficiales de Red Hat. Se utiliza el parámetro `--aid`.

```
$ rpm -ivh --aid libjpeg-6.2.0-738.i586.rpm
```

8. Actualizaciones automatizadas

Cada distribución facilita ahora una herramienta de actualización interactiva o automatizada. La distribución openSUSE ofrece **YOU** (*Yast Online Update*), la Red Hat ofrece **up2date.** Al ser de pago la versión RHEL de Red Hat, el acceso a las actualizaciones depende de un número de licencia y de una inscripción a **RHN** (*Red Hat Network*). Las versiones derivadas, como CentOS, no se pueden actualizar mediante RHN, pero ofrecen su propio sitio remoto de actualización. Un gran número de distribuciones recientes utilizan hoy en día PackageKit, disponible en la siguiente dirección:http://www.packagekit.org/

YUM

YUM es a los archivos **rpm** lo que APT a los archivos **dpkg**: un programa de gestión de paquetes. Recupera los paquetes dentro de los repositorios y gestiona las dependencias en lugar de usted. YUM significa *Yellow dog Updater Modified*. Se utiliza principalmente en las distribuciones Red Hat (versiones Enterprise) y Fedora, pero se puede utilizar en cualquier distribución de tipo RPM si los repositorios asociados lo soportan.

Los comandos y ejemplos siguientes se basan en un servidor Red Hat Enterprise Linux 5. El archivo de configuración es **`/etc/yum.conf`**.

1. Configuración de los repositorios

Los repositorios se ubican o en el archivo de configuración principal, o en el directorio **`/etc/yum.repos.d`**. El formato es el siguiente:

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
$releasever/Everything/$basearch/os/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-
$releasever&arch=$basearch
enabled=1
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
skip_if_unavailable=False
```

El repositorio se llama (nombre corto) `fedora`.

- **`name`**: el nombre largo del repositorio, detallado.
- **`baseurl`**: la URL del repositorio.
- **`gpgcheck`**: requiere una verificación de la firma GPG del repositorio.
- **`enabled`**: si no está definido o en 1, el repositorio está activo.
- **`gpgkey`**: ruta de la clave pública GPG.

Las URL de los repositorios pueden ser locales (file:///) o remotas (http://, https:// o ftp://). Deben apuntar a un directorio que contiene la información de repositorios que se encuentran en la carpeta `repodata`.

El archivo de configuración puede hacer uso de variables, como en este caso $basearch (arquitectura de 32 o 64 bits) o $releasever (la versión de la distribución). El comando yum rellena automáticamente estas variables según la instalación. Por supuesto, se les puede asignar los valores que usted crea oportunos.

Teniendo en cuenta los valores por defecto, se puede declarar un simple repositorio de la manera siguiente:

```
[updates-rhel7]
name=UPDATES-RHEL7
baseurl=ftp://ftp.server.com/RPMS.rhel7_updates_x86
```

Sin embargo, tenga cuidado, ya que la configuración de YUM puede modificar los valores por defecto. La sección [main] de **`/etc/yum.conf`** puede contener la línea:

```
gpgcheck=1
```

En este caso tendrá que modificar el valor `gpgcheck` a 0 en los repositorios que no necesitan firma.

2. Utilización de los repositorios

a. Refrescar la caché

Con cada comando, YUM intenta refrescar sus datos si se ha superado el plazo de expiración. Se puede reducir o ampliar este plazo con la modificación de la línea `metadata_expire` del archivo de configuración.

```
metadata_expire=1h
```

Puede forzar la actualización de la caché con el parámetro `makecache`:

```
[root@slyserver ]# yum makecache
Complementos cargados: langpacks, refresh-packagekit
fedora/20/x86_64/metalink                              |  34 kB  00:00
fedora                                                 | 3.8 kB  00:00
updates/20/x86_64/metalink                             |  30 kB  00:00
updates                                                | 4.9 kB  00:00
(1/9): fedora/20/x86_64/group_gz                       | 394 kB  00:01
(2/9): updates/20/x86_64/group_gz                      | 394 kB  00:03
(3/9): updates/20/x86_64/prestodelta                   | 2.4 MB  00:07
(4/9): fedora/20/x86_64/other_db                       | 8.7 MB  00:17
(5/9): updates/20/x86_64/primary_db                    |  11 MB  00:21
(6/9): updates/20/x86_64/other_db                      | 6.3 MB  00:15
(7/9): fedora/20/x86_64/primary_db                     |  18 MB  00:33
(8/9): fedora/20/x86_64/filelists_db                   |  27 MB  00:45
(9/9): updates/20/x86_64/filelists_db                  |  17 MB  00:44
(1/2): updates/20/x86_64/pkgtags                       | 1.0 MB  00:00
(2/2): updates/20/x86_64/updateinfo                    | 1.2 MB  00:01
Sa ha creado el caché de metadatos
```

Otra posibilidad consiste en forzar la supresión de la caché para que se vuelva a construir de manera automática con el próximo comando YUM:

```
root@slyserver ~]# yum clean all
Cleaning up Everything
```

b. Listar los paquetes

El parámetro **`list`** permite listar los paquetes. Todos se listan por defecto. Puede precisar una lista de paquetes o facilitar caracteres comodines. Hay varias opciones disponibles:

- **all**: es el caso por defecto: se listan primero los paquetes instalados y luego los disponibles para la instalación.
- **available**: los paquetes disponibles para instalación.
- **updates**: los paquetes que se pueden actualizar.
- **installed**: los paquetes actualizados.
- **obsoletes**: los paquetes del sistema obsoletos debido a las versiones superiores disponibles.
- **recent**: los últimos paquetes añadidos en los repositorios.

- **repolist**: los repositorios configurados en yum.conf o yum.repos.d.

El comando siguiente lista los núcleos disponibles:

```
[root@slyserver ~]# yum list available kernel\*

Paquetes disponibles
kernel.x86_64                              3.17.2-200.fc20          updates
kernel-debug.x86_64                        3.17.2-200.fc20          updates
kernel-debug-devel.x86_64                  3.17.2-200.fc20          updates
kernel-debug-modules-extra.x86_64          3.17.2-200.fc20          updates
kernel-devel.x86_64                        3.17.2-200.fc20          updates
kernel-doc.noarch                          3.17.2-200.fc20          updates
kernel-headers.x86_64                      3.17.2-200.fc20          updates
kernel-modules-extra.x86_64                3.17.2-200.fc20          updates
kernel-tools.x86_64                        3.17.2-200.fc20          updates
kernel-tools-libs.i686                     3.17.2-200.fc20          updates
kernel-tools-libs.x86_64                   3.17.2-200.fc20          updates
kernel-tools-libs-devel.i686               3.17.2-200.fc20          updates
kernel-tools-libs-devel.x86_64             3.17.2-200.fc20          updates
kernelshark.x86_64                         2.2.1-3.fc20             fedora
```

El parámetro `info` devuelve la información detallada de un paquete. Equivale al parámetro `-i` del comando `rpm`. Así, para el paquete `mc`, obtendrá algo parecido a lo siguiente:

```
[root@ slyserver ~]# yum info mc
Paquetes disponibles
Nombre        : mc
Arquitectura  : x86_64
Período       : 1
Versión       : 4.8.11
Lanzamiento   : 1.fc20
Tamaño        : 1.8 M
Repositorio   : updates/20/x86_64
Resumen       : User-friendly text console file manager and visual shell
URL           : http://www.midnight-commander.org/
Licencia      : GPLv3+
Descripción   : Midnight Commander is a visual shell much like a file manager, only
              : with many more features. It is a text mode application, but it also
              : includes mouse support. Midnight Commander's best features are its
              : ability to FTP, view tar and zip files, and to poke into RPMs for
              : specific files.
```

c. Instalar paquetes

Pase el parámetro `install`, seguido de los nombres de paquetes que va a instalar.

A continuación tenemos un ejemplo de instalación del paquete `mc`:

```
[root@slyserver etc]# yum install mc
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete mc.x86_64 1:4.8.11-1.fc20 debe ser instalado
--> Procesando dependencias: libgpm.so.2()(64bit) para el paquete: 1:mc-4.8.11-
.fc20.x86_64
--> Ejecutando prueba de transacción
---> Paquete gpm-libs.x86_64 0:1.20.7-3.fc20 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

================================================================================
 Package          Arquitectura   Versión                  Repositorio     Tamaño
```

```
================================================================================
Instalando:
 mc                x86_64         1:4.8.11-1.fc20          updates         1.8 M
Instalando para las dependencias:
 gpm-libs          x86_64         1.20.7-3.fc20            fedora           32 k

Resumen de la transacción
================================================================================
Instalar  1 Paquete (+1 Paquete dependiente)

Tamaño total de la descarga: 1.8 M
Tamaño instalado: 6.1 M
Is this ok [y/d/N]: y
Downloading packages:
(1/2): gpm-libs-1.20.7-3.fc20.x86_64.rpm                   |  32 kB   00:00
(2/2): mc-4.8.11-1.fc20.x86_64.rpm                         | 1.8 MB   00:04
--------------------------------------------------------------------------------
Total                                          391 kB/s | 1.8 MB     00:04

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando    : gpm-libs-1.20.7-3.fc20.x86_64                            1/2
  Instalando    : 1:mc-4.8.11-1.fc20.x86_64                                2/2
  Comprobando   : gpm-libs-1.20.7-3.fc20.x86_64                            1/2
  Comprobando   : 1:mc-4.8.11-1.fc20.x86_64                                2/2

Instalado:
  mc.x86_64 1:4.8.11-1.fc20

Dependencia(s) instalada(s):
  gpm-libs.x86_64 0:1.20.7-3.fc20

¡Listo!
```

d. Actualizaciones

Compruebe la publicación de actualizaciones con el parámetro `check-update`:

```
[root@slyserver /etc]# yum check-update
```

Si no devuelve nada quiere decir que no hay publicada ninguna actualización.

Tiene dos posibilidades para instalar las actualizaciones:

- **update**: actualización de un paquete o de todos si no se especifica ningún paquete.
- **upgrade**: actualización completa de la distribución: se sustituyen los paquetes considerados como obsoletos con los de la última versión disponible.

En algunos casos, para el núcleo por ejemplo, tendrá que evitar durante una actualización instalar automáticamente algunos paquetes. En este caso, utilice el parámetro `--exclude`:

```
[root@slyserver etc]# yum -exclude=kernel\* update
```

Para convertir esta exclusión en algo permanente, póngala en el archivo de configuración añadiendo una línea como la siguiente:

```
exclude=php* kernel*
```

e. Buscar un paquete

Utilice el parámetro `search`, seguido del paquete o de los paquetes que debe buscar en los

repositorios. Se autorizan los caracteres comodines.

```
[root@slyserver ~]# yum search fortune
Complementos cargados:langpacks, refresh-packagekit
============================== N/S matched: fortune
==============================
fortune-mod.x86_64 : A program which will display a fortune
fortune-firefly.noarch : Quotes from the TV series "Firefly"

  Nombre y resumen que coinciden con  y sólo , use "buscar todo" para todo.
```

f. Suprimir un paquete

Para suprimir un paquete, utilice el parámetro `remove`:

```
[root@slyserver ~]# yum remove mc

Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete mc.x86_64 1:4.8.11-1.fc20 debe ser eliminado
--> Resolución de dependencias finalizada

Dependencias resueltas

================================================================================
 Package     Arquitectura    Versión                     Repositorio       Tamaño
================================================================================
Eliminando:
 mc          x86_64          1:4.8.11-1.fc20             @updates          6.1 M

Resumen de la transacción
================================================================================
Eliminar  1 Paquete

Tamaño instalado: 6.1 M
Está de acuerdo [s/N]:s
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Eliminando    : 1:mc-4.8.11-1.fc20.x86_64                                  1/1
  Comprobando   : 1:mc-4.8.11-1.fc20.x86_64                                  1/1

Eliminado(s):
  mc.x86_64 1:4.8.11-1.fc20

¡Listo!
```

Debian Package

1. dpkg: el gestor de paquetes Debian

El comando **dpkg** es el equivalente de rpm para las distribuciones Debian y derivados, entre las cuales se encuentra Ubuntu. Hace lo mismo o casi que **rpm**. Los paquetes Debian llevan una extensión .deb para reconocerlos y disponen de la misma información y mismos medios que un paquete **rpm**. El comando **dpkg** es el encargado de la instalación, la creación, la supresión y la gestión de los paquetes Debian.

Se suele colocar la base de datos **dpkg** en **/var/lib/dpkg**. Los archivos de este directorio tienen formato de texto. Sin embargo, no edite los archivos manualmente. El archivo **/var/lib/dpkg/status** contiene la totalidad de los paquetes conocidos por **dpkg** con su estado.

```
# grep ^Package: /var/lib/dpkg/status | grep glibc
Package: devhelp-book-glibc
Package: libg++2.8.1.3-glibc2.2
Package: glibc-doc
Package: libstdc++2.10-glibc2.2
```

2. Instalación, actualización y supresión

La opción `-i`, o `--install`, instala el paquete o los paquetes que se pasan como argumento.

```
# dpkg -i mipaquete.deb
```

Observe que, al igual que rpm, `dpkg` no gestiona las dependencias. Si faltan dependencias, el comando le informará. En este caso, tendrá que instalar las dependencias de la misma manera antes de instalar su paquete.

Puede pedir la instalación de todos los paquetes presentes en el árbol con el parámetro `-R`, la inicial de Recursividad. En este caso, indique como argumento un nombre de directorio y se instalarán todos los paquetes presentes en el directorio, así como sus subdirectorios.

```
# dpkg -R directorio
```

Se actualiza de la misma manera que se instala, con **-i**. Si instala un paquete ya presente, `dpkg`efectúa su actualización. Así, una instalación o una actualización respeta la metodología siguiente:

- Extracción de los archivos de control del nuevo paquete.
- Cuando ya se instaló una antigua versión del mismo paquete, se ejecuta el script presupresión del antiguo paquete.
- Ejecución del script de preinstalación si se facilita con el paquete.
- Desempaquetado de nuevos archivos y copia de seguridad de los antiguos para poder restaurarlos en caso de problema.
- Cuando ya se ha instalado una antigua versión del paquete, se ejecuta el script postsupresión del antiguo paquete.
- Configuración del paquete.

No existe equivalencia con el modo freshen (-F) de rpm. Si desea actualizar un paquete únicamente en caso de que ya esté instalado, debe comprobar primero si lo está. Si es el caso, el paquete comienza por ii en la lista:

Para

```
# (dpkg -l zip | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
PRESENT
# (dpkg -l slapd | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
ABSENT
```

actualizar un paquete sólo en el caso de que ya esté presente, utilice este tipo de línea de comandos o un

equivalente de su invención:

```
# (dpkg -l zip | grep ^ii >/dev/null) && dpkg -l zip.deb
```

Observe que lo más fácil para actualizar sus paquetes ya presentes es crear un repositorio APT que contenga sus actualizaciones y ejecutar un apt-get upgrade desde el cliente.

La supresión de un paquete se lleva a cabo con el parámetro `-r` (en minúscula). Esta vez, también le corresponde a usted gestionar las dependencias.

La supresión de un paquete sigue las etapas siguientes:

- Ejecución del script de presupresión.
- Supresión de los archivos instalados.
- Ejecución del script de postsupresión.

```
# dpkg -r zip
```

Se suprime todo salvo los archivos de configuración para evitar una nueva configuración de la herramienta si vuelve a instalarla. Para suprimir todo, incluyendo estos archivos, indique el parámetro `-P` (purga).

```
# dpkg -P apache
```

Cuidado: ¡no confunda los parámetros `-r` y `-R`!

Si sustituye el nombre del paquete por los parámetros `-a` o `--pending`, se borran los paquetes no instalados (no desempaquetados) pero que están marcados en la base de datos para purgarlos o suprimirlos.

La utilización de las opciones `--force-all` y `--purge` permite forzar la desinstalación del paquete y suprimir los archivos de configuración asociados.

```
# dpkg --force-all --purge nombre_del_paquete
```

3. Consultas con dpkg

a. Listar los paquetes

Puede listar todos los paquetes Debian conocidos del sistema con el parámetro `-l`:

```
# dpkg -l
...
ii  mount             2.20.1-5.3        amd64  Tools for mounting and manipulating filesystems
ii  mousetweaks       3.4.2-1           amd64  mouse accessibility enhancements for the GNOME
                                                desktop
ii  mscompress        0.3-4             amd64  Microsoft "compress.exe/expand.exe" compatible
                                                (de)compressor
ii  multiarch-support 2.13-38+deb7u6    amd64  Transitional package to ensure multiarch
                                                compatibility
ii  mutt              1.5.21-6.2+deb7u2 amd64  text-based mailreader supporting MIME, GPG, PGP
                                                and threading
ii  mutter-common     3.4.1-5           all    shared files for the Mutter window manager
ii  myspell-es        1.11-4            all    Spanish dictionary for myspell
ii  mythes-en-us      1:3.3.0-4         all    English Thesaurus for LibreOffice/OpenOffice.org
ii  nano              2.2.6-1+b1        amd64  small, friendly text editor inspired by Pico
ii  nautilus          3.4.2-1+build1    amd64  file manager and graphical shell for GNOME
ii  nautilus-data     3.4.2-1+build1    all    data files for nautilus
ii  nautilus-sendto   3.0.3-2+b1        amd64  integrates Evolution and Pidgin into the Nautilus
                                                file manag
```

```
...
```

Puede indicar un motivo particular:

```
# dpkg -l "apt*"
...
ii  apt                0.9.7.9+deb7u6 amd64  commandline package manager
un  apt-doc            <ninguna>             (no hay ninguna descripción disponible)
un  apt-forktracer     <ninguna>             (no hay ninguna descripción disponible)
ii  apt-listchanges    2.85.11        all    package change history notification tool
un  apt-p2p            <ninguna>             (no hay ninguna descripción disponible)
ii  apt-utils          0.9.7.9+deb7u6 amd64  package managment related utility programs
ii  apt-xapian-index   0.45             all  maintenance and search tools
for a Xapian index of De
```

Si su consola es demasiado pequeña para visualizar los nombres de los paquetes (segunda columna), puede valerse del truco siguiente:

```
# COLUMNS=160 dpkg -l "kernel*" | grep ^ii | awk '{print $2}'
linux-base
linux-headers-3.2.0-4-amd64
linux-headers-3.2.0-4-common
linux-headers-amd64
linux-image-3.2.0-4-amd64
linux-image-amd64
linux-kbuild-3.2
linux-libc-dev:amd64
```

Otro método consiste en emplear la opción **`--get-selections`**:

```
# dpkg --get-selections | grep ^linux
linux-base                                  install
linux-headers-3.2.0-4-amd64                 install
linux-headers-3.2.0-4-common                install
linux-headers-amd64                         install
linux-image-3.2.0-4-amd64                   install
linux-image-amd64                           install
linux-kbuild-3.2                            install
linux-libc-dev:amd64                        install
```

b. Encontrar un paquete que contiene un archivo

El parámetro `-S` seguido del nombre de un archivo (su ruta) permite encontrar el paquete de origen.

```
# dpkg -S /usr/bin/basename
coreutils: /usr/bin/basename
```

c. Listar el contenido de un paquete

El parámetro `-L` lista el contenido del paquete o de los paquetes indicados:

```
# dpkg -L coreutils | grep bin
/bin
/bin/mkdir
/bin/mv
/bin/true
/bin/mknod
/bin/sleep
/bin/touch
/bin/chgrp
/bin/uname
/bin/echo
/bin/sync
```

```
/bin/ln
/bin/date
/bin/dir
/bin/readlink
...
```

4. Convertir paquetes

La herramienta **alien** permite convertir paquetes RPM en DPKG y viceversa. Algunos paquetes sólo se facilitan para un sistema u otro. Cuando se suministra un producto de software sólo bajo una forma, es molesto tener que instalarlo de todas maneras en otra plataforma Linux.

A continuación, ponemos el ejemplo de un paquete, el cliente Networker, proporcionado solamente para Red Hat. Es posible convertirlo al formato dpkg con Alien.

El parámetro por defecto `-d` convierte del rpm al dpkg:

```
# alien -d lgtoclnt-7.4-1.i686.rpm
Warning: Skipping conversion of scripts in package lgtoclnt: postinst postrm
preinst prerm
Warning: Use the --scripts parameter to include the scripts.
lgtoclnt_7.4-2_i386.deb generated
```

Como se ha indicado ya, la conversión por defecto va a comprobar las dependencias, pero no va a incluir los scripts de pre y postinstalación. En ese caso debe precisar el parámetro `--scripts`.

El

```
# alien --scripts -d lgtoclnt-7.4-1.i686.rpm
lgtoclnt_7.4-2_i386.deb generated
# ls -l *.deb
-rw-r--r-- 1 root root 29471546 2008-05-09 14:45 lgtoclnt_7.4-2_i386.deb
```

resultado es el siguiente:

```
# dpkg -I lgtoclnt_7.4-2_i386.deb
 nuevo paquete Debian, versión 2.0.
 tamaño 29471546 bytes: archivo de control = 4498 bytes.
     923 bytes,    18 líneas      control
    3142 bytes,    57 líneas      md5sums
    4014 bytes,   148 líneas   *  postinst             #!/bin/sh
    1362 bytes,    35 líneas   *  postrm               #!/bin/sh
     317 bytes,    11 líneas   *  preinst              #!/bin/sh
    1828 bytes,    52 líneas   *  prerm                #!/bin/sh
      61 bytes,     3 líneas      shlibs
 Package: lgtoclnt
 Version: 7.4-2
 Section: alien
 Priority: extra
 Arquitectura: i386
 Depends: libc6 (>= 2.3.6-6), libgl1-mesa-glx | libgl1, libice6 (>=
1:1.0.0), libncurses5 (>= 5.4-5), libsm6, libx11-6, libxext6,
libxmu6, libxp6, libxrender1, libxt6
 Installed-Size: 71632
 Maintainer: root <root@s64p17bib76.dsit.renfe.es>
 Description: NetWorker Client
  EMC NetWorker protects the critical business data of more than 10,000
...
  and the smallest satellite branch offices.
  .
  (Converted from a rpm package by alien version 8.64.)
```

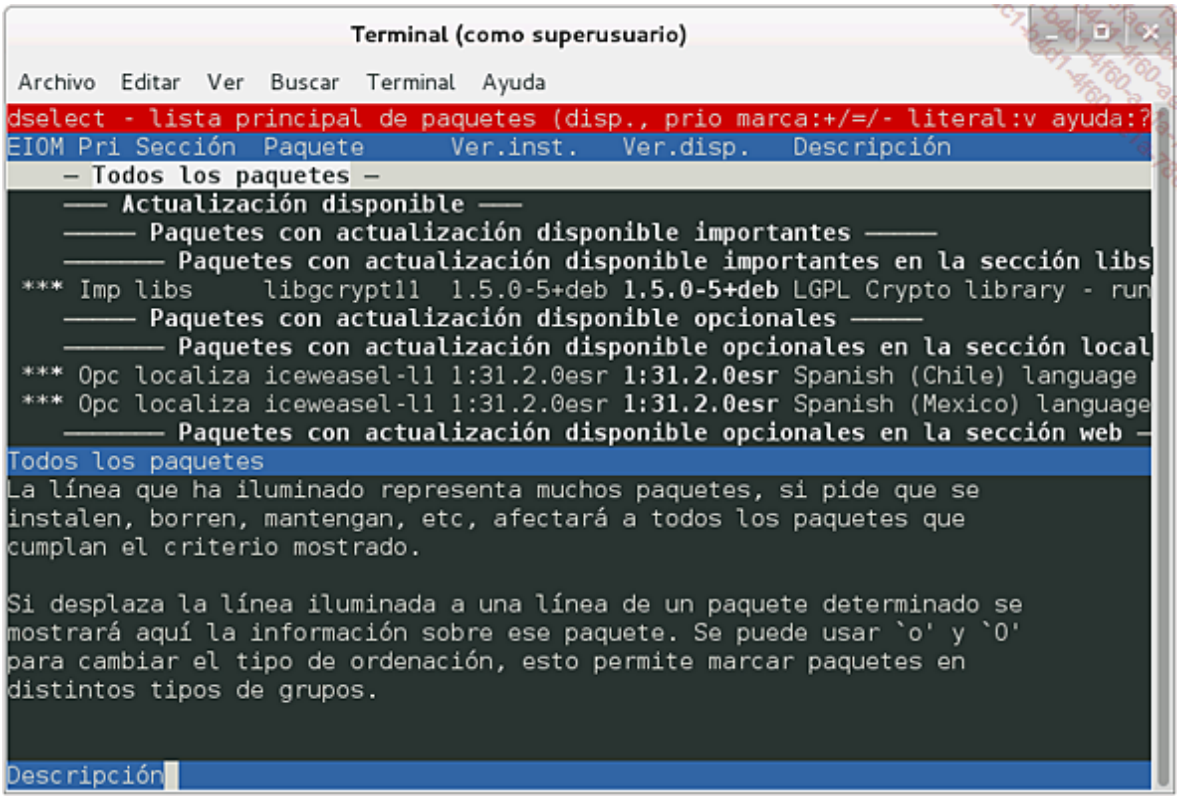
5. La herramienta dselect

La herramienta **dselect** es un frontend (como APT) para dpkg, que gestiona las dependencias y los conflictos. Históricamente fue la primera. Sin embargo, su sustituto APT tiene mejor calidad.

El manual de dselect indica de manera clara que hoy en día la herramienta ha caído en desuso.

- La interfaz es confusa.
- Hay poco mantenimiento.
- El manual es incompleto.
- No hay filtro.
- Los accesos ya no son estándares.

Por todos estos motivos es preferible utilizar APT.



Se recomienda no seguir utilizando dselect.

Gestor APT

1. Fundamentos

Ya sea con rpm o dpkg, el problema sigue siendo el mismo: estas dos herramientas detectan las dependencias de los paquetes para autorizar o no su instalación, pero no las resuelven. Dicho de otro modo, si una dependencia de un paquete no está, no se instalará salvo si se resuelve la dependencia:

- o instalando previamente los paquetes que faltan,
- o indicando en la misma línea la ruta de estos mismos paquetes.

Asimismo, en el momento de la actualización se presenta un problema con los archivos de configuración. ¿Qué se debe hacer?

APT permite resolver estos problemas gestionando las dependencias en lugar de usted. **APT**significa *Advanced Packaging Tool*. En vez de especificar un paquete (local o remoto), se encarga de los repositorios de paquetes situados en un CD, un DVD, en un directorio local, en una fuente remota en Internet (ftp, http), etc.

Un repositorio contiene un conjunto de paquetes que dependen o bien los unos de los otros, o bien de otros paquetes procedentes de otros repositorios. APT puede gestionar varios repositorios en varios sitios. Se las arregla solo: cuando usted instala un paquete, también instala sus dependencias (si las encuentra).

2. Los repositorios

a. Configuración

Los repositorios quedan fijados en el archivo **`/etc/apt/sources.list`**. El siguiente archivo proviene de una instalación Debian Etch en la cual se añadieron los repositorios contrib y non-free.

```
$ cat /etc/apt/sources.list
deb http://ftp.es.debian.org/debian/ wheezy main
deb-src http://ftp.es.debian.org/debian/ wheezy main

deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main

# wheezy-updates, previously known as 'volatile'
deb http://ftp.es.debian.org/debian/ wheezy-updates main
deb-src http://ftp.es.debian.org/debian/ wheezy-updates main
```

Los repositorios de software se ubican físicamente en el árbol de directorios del servidor. El comando **genbasedir** permite crear un repositorio. La sintaxis de una línea del archivo**`sources.list`** es la siguiente:

```
deb uri distribución componente1 componente2 ...
```

- `uri` es la ruta hacia la raíz del repositorio o de los repositorios. Puede ser una URL de tipo http o ftp, pero también una ruta local (file), un CD-Rom o DVD-Rom (CDrom), una ruta ssh, etc.
- La distribución es, como su nombre indica, el nombre de la distribución Debian. Aquí es**wheezy**, pero es posible especificar otras versiones de la distribución (testing, etc.) para poder recuperar paquetes de otros repositorios, más recientes por ejemplo. Se puede precisar la arquitectura. Si no es el caso, APT se las arregla solo para añadir el sufijo

necesario.

- Los componentes son los nombres de los repositorios para la distribución dada.

En la práctica, la uri, la distribución y los componentes permiten reconstituir la URL completa de acceso al repositorio.

- Diríjase a http://ftp.es.debian.org/debian/.
- Haga clic en el archivo llamado **dists**. Contiene la lista de las distribuciones Debian.
- En **dists**, haga clic en **wheezy**, el nombre de la distribución actual.
- En **squeeze**, encontrará las carpetas **contrib**, **main** y **non-free**.

La línea `deb http://ftp.es.debian.org/debian/ wheezy main contrib non-free` corresponde por lo tanto a las URL:

- http://ftp.es.debian.org/debian/dists/wheezy/main
- http://ftp.es.debian.org/debian/dists/wheezy/contrib
- http://ftp.es.debian.org/debian/dists/wheezy/non-free

Si continúa, por ejemplo entrando en **main**, encontrará una serie de carpetas con sufijos según la arquitectura de su instalación de Linux. La máquina VMWare de prueba es de tipo amd64. Por lo tanto, se buscarán los paquetes binarios en **binary-amd64**.

No se sorprenda de no encontrar paquetes en este último directorio, sino archivos:

- **`Release`**: descripción del repositorio.
- **`Packages.gz`**: índice de los paquetes del repositorio en el formato gzip.
- **`Packages.bz2`**: lo mismo en el formato bzip2.

¿Dónde están realmente los paquetes? La respuesta está en los archivos Packages.*. Las primeras líneas de uno de estos archivos tienen este aspecto:

```
Package: abiword-common
Source: abiword
Version: 2.9.2+svn20120603-8
Installed-Size: 10960
Maintainer: Dmitry Smirnov <onlyjob@member.fsf.org>
Architecture: all
Description: efficient, featureful word processor with collaboration -- common files
Homepage: http://www.abisource.com/
Description-md5: 168081fc8391dc5eb8f29d63bb588273
Tag: interface::x11, role::app-data, uitoolkit::gtk, use::editing,
 use::text-formatting, works-with-format::html, works-with-format::tex,
 works-with::text, x11::application
Section: editors
Priority: optional
Filename: pool/main/a/abiword/abiword-common_2.9.2+svn20120603-8_all.deb
Size: 2292710
```

La línea **Filename** le indica que el archivo está en **`pool/main/a/abiword/`** desde el uri, y por lo tanto accesible desde la URL http://ftp.es.debian.org/debian/pool/main/a/abiword/. Observe también la descripción de los paquetes, las líneas Depends y Recommends que permiten a APT resolver las dependencias. Este archivo se parece mucho, y con motivos, al archivo **`status`** de la base dpkg local.

b. Actualización de la base de datos

Una vez configurados sus repositorios, debe actualizar la base de datos local de APT con el

comando **apt-get** y la opción **update**.

```
root@debian-LP1:~# apt-get update
Des:1 http://ftp.es.debian.org wheezy Release.gpg [1.655 B]
Des:2 http://ftp.es.debian.org wheezy-updates Release.gpg [836 B]
Des:3 http://ftp.es.debian.org wheezy Release [168 kB]
Obj http://security.debian.org wheezy/updates Release.gpg
Obj http://security.debian.org wheezy/updates Release
Obj http://security.debian.org wheezy/updates/main Sources
Des:4 http://ftp.es.debian.org wheezy-updates Release [124 kB]
Obj http://security.debian.org wheezy/updates/main amd64 Packages
Obj http://security.debian.org wheezy/updates/main Translation-en
Des:5 http://ftp.es.debian.org wheezy/main Sources [5.955 kB]
Des:6 http://ftp.es.debian.org wheezy/main amd64 Packages [5.841 kB]
Des:7 http://ftp.es.debian.org wheezy/main Translation-es [349 kB]
Des:8 http://ftp.es.debian.org wheezy/main Translation-en [3.846 kB]
Des:9 http://ftp.es.debian.org wheezy-updates/main Sources [14 B]
Des:10 http://ftp.es.debian.org wheezy-updates/main amd64 Packages/DiffIndex [1.609 B]
Des:11 http://ftp.es.debian.org wheezy-updates/main Translation-en/DiffIndex [1.057 B]
Descargados 16,3 MB en 19seg. (816 kB/s)
Leyendo lista de paquetes... Hecho
```

3. Actualización de la distribución

Una vez actualizados los repositorios, puede actualizar en un solo comando todos los paquetes instalados en su distribución: APT verifica si unos paquetes más recientes están disponibles en los repositorios. Se basa para ello en la base de datos local. Si no está actualizada, es posible que ya no se encuentran algunos de los paquetes más antiguos.

Ejecute el comando **apt-get** con la opción **upgrade**. APT le informa que se pueden actualizar ocho paquetes. Puede aceptar o rechazar. Si acepta, APT descarga estos paquetes y sus posibles dependencias, y los instala. El proceso puede ser más o menos largo según el número de actualizaciones y el tipo de soporte.

```
# apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando Árbol de dependencias
Leyendo la información de estado... Hecho
Se actualizarán los siguientes paquetes:
  libjasper-runtime libjasper1 libsoup-gnome2.4-1 libsoup2.4-1 x11-common
  xbase-clients xorg xserver-xorg xserver-xorg-input-all
  xserver-xorg-video-all
10 actualizados, 0 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 867 kB de archivos.
Se utilizarán 131 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://security.debian.org/ squeeze/updates/main libjasper1
amd64 1.900.1-7+squeeze1 [157 kB]
Des:2 http://security.debian.org/ squeeze/updates/main libjasper-runtime
amd64 1.900.1-7+squeeze1 [26,2 kB]
Des:3 http://security.debian.org/ squeeze/updates/main libsoup2.4-1
amd64 2.30.2-1+squeeze1 [176 kB]
Des:4 http://security.debian.org/ squeeze/updates/main libsoup-gnome2.4-1
amd64 2.30.2-1+squeeze1 [41,1 kB]
Des:5 http://security.debian.org/ squeeze/updates/main x11-common all
1:7.5+8+squeeze1 [279 kB]
Des:6 http://security.debian.org/ squeeze/updates/main xbase-clients all
1:7.5+8+squeeze1 [38,2 kB]
Des:7 http://security.debian.org/ squeeze/updates/main xserver-xorg-video-all
amd64 1:7.5+8+squeeze1 [32,0 kB]
Des:8 http://security.debian.org/ squeeze/updates/main xserver-xorg-input-all
amd64 1:7.5+8+squeeze1 [31,8 kB]
Des:9 http://security.debian.org/ squeeze/updates/main xserver-xorg
```

```
amd64 1:7.5+8+squeeze1 [53,3 kB]
Des:10 http://security.debian.org/ squeeze/updates/main xorg
amd64 1:7.5+8+squeeze1 [32,5 kB]
Descargados 867 kB en 4seg. (201 kB/s)
Leyendo lista de cambios... Hecho.
Preconfigurando paquetes ...
(Leyendo la base de datos ... 195159 ficheros o directorios instalados actualmente.)
Preparando para reemplazar libjasper1 1.900.1-7+b1
(usando .../libjasper1_1.900.1-7+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de libjasper1 ...
Preparando para reemplazar libjasper-runtime 1.900.1-7+b1
(usando .../libjasper-runtime_1.900.1-7+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de libjasper-runtime ...
Preparando para reemplazar libsoup2.4-1 2.30.2-1
(usando .../libsoup2.4-1_2.30.2-1+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de libsoup2.4-1 ...
Preparando para reemplazar libsoup-gnome2.4-1 2.30.2-1
(usando .../libsoup-gnome2.4-1_2.30.2-1+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de libsoup-gnome2.4-1 ...
Preparando para reemplazar x11-common 1:7.5+8
(usando .../x11-common_1%3a7.5+8+squeeze1_all.deb) ...
Desempaquetando el reemplazo de x11-common ...
Preparando para reemplazar xbase-clients 1:7.5+8
(usando .../xbase-clients_1%3a7.5+8+squeeze1_all.deb) ...
Desempaquetando el reemplazo de xbase-clients ...
Preparando para reemplazar xserver-xorg-video-all 1:7.5+8
(usando .../xserver-xorg-video-all_1%3a7.5+8+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de xserver-xorg-video-all ...
Preparando para reemplazar xserver-xorg-input-all 1:7.5+8
(usando .../xserver-xorg-input-all_1%3a7.5+8+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de xserver-xorg-input-all ...
Preparando para reemplazar xserver-xorg 1:7.5+8
(usando .../xserver-xorg_1%3a7.5+8+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de xserver-xorg ...
Preparando para reemplazar xorg 1:7.5+8
(usando .../xorg_1%3a7.5+8+squeeze1_amd64.deb) ...
Desempaquetando el reemplazo de xorg ...
Procesando disparadores para man-db ...
Configurando libjasper1 (1.900.1-7+squeeze1) ...
Configurando libjasper-runtime (1.900.1-7+squeeze1) ...
Configurando libsoup2.4-1 (2.30.2-1+squeeze1) ...
Configurando libsoup-gnome2.4-1 (2.30.2-1+squeeze1) ...
Configurando x11-common (1:7.5+8+squeeze1) ...
Configurando xbase-clients (1:7.5+8+squeeze1) ...
Configurando xserver-xorg-video-all (1:7.5+8+squeeze1) ...
Configurando xserver-xorg-input-all (1:7.5+8+squeeze1) ...
Configurando xserver-xorg (1:7.5+8+squeeze1) ...
Configurando xorg (1:7.5+8+squeeze1) ...
```

Otra posibilidad consiste en realizar una actualización a fondo (llamada actualización remota). ATP guarda una cierta coherencia en los paquetes durante la actualización, en particular en cuanto a la versión de la distribución. Puede especificar varias distribuciones Debian en sus repositorios. Pero incluso si una distribución es más reciente, un simple upgrade no va a transformar la suya en la última. Puede pedir a APT que fuerce la actualización hacia la nueva distribución con un **dist-upgrade**.

Para los objetivos de este libro, se han añadido los repositorios de la versión de prueba de Debian llamada testing:

```
## testing
deb http://ftp.es.debian.org/debian/ testing main contrib non-free
deb-src http://ftp.es.debian.org/debian/ testing main contrib non-free
# security testing
deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

Efectúe una actualización de la base. Observe que para ello ha hecho falta ampliar la caché de APT y limpiar la base:

```
# apt-get update
echo 'APT::Cache-Limit "141943904";' > /etc/apt/apt.conf.d/00Cache
# apt-get clean
# apt-get update
```

luego

```
$ apt-get dist-upgrade
Lectura de las listas de paquetes...
Construcción del arbol de las dependencias...
...
731 actualizados, 228 nuevamente instalados, 23 a quitar y 0 no actualizados.
Es necesario coger 858 Mb en las carpetas.
Después del desempaquetado, se utilizarán 623 Mb de espacio en disco adicional.
¿Desea proseguir [S/n]? S
...
```

¡Mucha suerte!

4. Buscar e instalar un paquete individual

El comando **apt-cache** permite buscar un paquete por su nombre o su comentario en la base de datos local APT.

```
root@debian:~# apt-cache search torrent
vuze - Multimedia BitTorrent client
bitstormlite - BitTorrent Client based on C++/Gtk+2.0
bittornado-gui - bittorrent client with GUI interface
bittornado - bittorrent client (and tracker) with console and curses interfaces
bittorrent-gui - Original BitTorrent client - GUI tools
bittorrent - Original BitTorent client - console tools
ctorrent - BitTorrent Client written in C++
debtorrent - bittorrent proxy for downloading Debian packages
```

El comando **apt-get install** xxx instala el paquete xxx:

```
root@debian:~# apt-get install vim-gtk
Leyendo lista de paquetes... Hecho
Creando Árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libruby1.9.1 libyaml-0-2 tcl8.5 vim-gui-common vim-runtime
Paquetes sugeridos:
  tcl-tclreadline cscope vim-doc
Se instalarán los siguientes paquetes NUEVOS:
  libruby1.9.1 libyaml-0-2 tcl8.5 vim-gtk vim-gui-common vim-runtime
0 actualizados, 6 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 11,9 MB de archivos.
Se utilizarán 42,2 MB de espacio de disco adicional después de esta operación.
¿Desea proseguir [S/n]? S
...
```

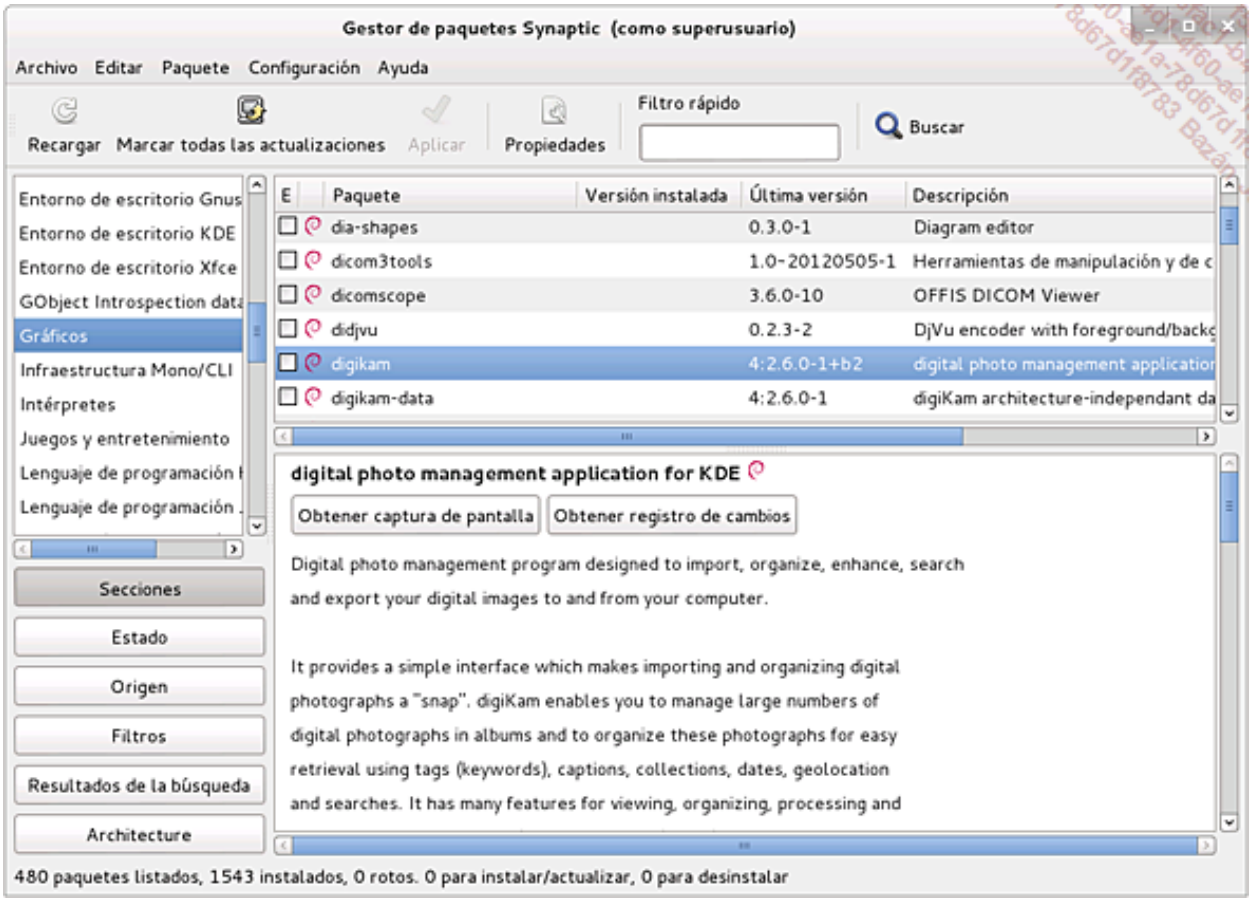
Dos opciones merecen recordarse:

- la **-s** para la simulación: APT indica lo que debería hacer, pero no lo hace.

- la **-f** para "fix-broken": APT intenta arreglar los problemas de dependencias como puede (añadido de paquetes).

5. Cliente gráfico

La herramienta synaptic es un front-end: una interfaz gráfica que llama a las funciones de APT. Soporta todas las operaciones propuestas por APT y, además, disfruta de una interfaz muy atractiva.



Synaptic es un front-end para APT.

Instalar desde las fuentes

1. Obtener las fuentes

A veces no es posible obtener un programa o una librería desde un paquete para su distribución. En este caso, queda la solución de compilar e instalar uno mismo el programa desde las fuentes.

Eso es posible para la mayoría de los programas en Linux gracias a las ventajas de los programas libres y de la licencia GPL tal como quedó definida en el primer capítulo. Cualquier programa libre va acompañado con sus fuentes. Por lo tanto, usted mismo puede volver a reconstruir el programa al compilarlo.

Es posible obtener un archivo fuente en diversos sitios web, como por ejemplo SourceForge. Suele ser un archivo comprimido en formato tgz (archivo tar comprimido con gzip) o tar.bz2 (archivo tar comprimido en formato bzip2). Contiene:

- el código fuente en forma de archivos **`.c`**, **`.h`**, **`.cpp`**, etc., según el lenguaje;
- a veces un archivo **`Makefile`** que permite automatizar la compilación del producto;
- a menudo un archivo **`.configure`** que permite generar el archivo Makefile en función de su instalación y de varias opciones.

2. Requisitos y dependencias

Para compilar su producto, tiene que respetar unos requisitos:

- presencia de la herramienta make;
- presencia del compilador o de los compiladores necesarios, en particular gcc;
- presencia de las dependencias: librerías, intérpretes, etc.

Este último punto es muy importante. Si falta una dependencia, se arriesga a encontrarse con varios problemas:

- no logrará preparar las fuentes para la compilación;
- la compilación generará errores;
- se compilará el programa pero con menos opciones;
- no se iniciará el binario resultante.

El comando **./configure** le proporcionará las dependencias que faltan y su versión si es posible. En este caso, puede instalarlas desde los paquetes de su distribución, o bien instalarlas desde las fuentes.

Cuando se compila desde las fuentes sin pasar por un gestor de paquetes, se pierde una parte de la gestión de las dependencias. Cuando instala paquetes que dependen de una versión de la herramienta instalada desde las fuentes, es posible, si las dependencias se basan en la existencia de un paquete y no de un archivo, que el gestor le impida instalar el paquete. Busque correctamente entre los repositorios, oficiales o no, si el programa existe en forma de paquete antes de volver a compilarlo o si existe un paquete fuente.

En cualquier caso, no hace falta que sea root para compilar el programa. Sin embargo, según el destino de la aplicación, tendrá que pasar a root para finalizar la instalación.

3. Ejemplo de instalación

Vamos a compilar e instalar el producto PDFedit que permite editar y crear archivos PDF.

- Descárguelo desde el vínculo siguiente. La versión probada es la versión 0.4.5:http://sourceforge.net/project/showfiles.php?group_id=177354

```
$ ls -l pdfedit.tar.bz2
-rw-r--r-- 1 seb seb 21742 dic 17 17:52 pdfedit.tar.bz2
```

- Descomprima el archivo:

```
$ tar xvjf pdfedit-0.4.5.tar.bz2
pdfedit-0.4.5/COPYING
pdfedit-0.4.5/Changelog
pdfedit-0.4.5/Makefile
pdfedit-0.4.5/Makefile.flags.in
pdfedit-0.4.5/Makefile.rules.in
pdfedit-0.4.5/README
pdfedit-0.4.5/config/
pdfedit-0.4.5/config/xpdf.m4
pdfedit-0.4.5/config/pdfedit-core-dev.m4
pdfedit-0.4.5/config/macro.m4
...
```

- Vaya a la carpeta **`pdfedit-0.4.5`** creada por la descompresión:

```
$ cd pdfedit-0.4.5/
$ ls -l
total 400
-rw-r--r--  1 seb seb   8880 may 11  2010 Changelog
drwxr-xr-x  2 seb seb   4096 may 11  2010 config
-rwxr-xr-x  1 seb seb 284547 may 11  2010 configure
-rw-r--r--  1 seb seb  20849 feb 23  2010 configure.in
-rw-r--r--  1 seb seb    969 feb 21  2008 COPYING
drwxr-xr-x  5 seb seb   4096 may 11  2010 doc
-rwxr-xr-x  1 seb seb   4275 may 11  2010 getversion
-rw-r--r--  1 seb seb   3451 nov 13  2008 Makefile
-rw-r--r--  1 seb seb   9351 feb 23  2010 Makefile.flags.in
-rw-r--r--  1 seb seb   1455 oct 21  2009 Makefile.rules.in
drwxr-xr-x 11 seb seb   4096 may 11  2010 projects
-rw-r--r--  1 seb seb  21822 feb 23  2010 README
-rw-r--r--  1 seb seb     15 may 11  2010 release_stamp
drwxr-xr-x 11 seb seb   4096 may 11  2010 src
drwxr-xr-x  3 seb seb   4096 may 11  2010 tools
-rw-r--r--  1 seb seb     62 oct 30  2008 version_exclude
```

- Observe la presencia del archivo configure que es ejecutable.

```
seb@debian:~/pdfedit/pdfedit-0.4.5$ ./configure --help
`configure' configures PDFedit 0.4.5 to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
```

```
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print `checking...' messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for `--cache-file=config.cache'
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR
[configure dir or `..']

Installation directories:
  --prefix=PREFIX         install architecture-independent files in PREFIX
...
```

Una opción importante de configure es **--prefix**. Define la ubicación de la instalación una vez compilado el producto. Por defecto, el programa se instala en **/usr/local/**.

- Ejecute **./configure** solo. Le informará de las dependencias que faltan en caso necesario.

```
root@debian:/home/seb/pdfedit/pdfedit-0.4.5# ./configure
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for gcc... gcc
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking for ranlib... ranlib
checking whether ln -s works... yes
checking how to run the C++ preprocessor... g++ -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking limits.h usability... yes
checking limits.h presence... yes
checking for limits.h... yes
checking for stdlib.h... (cached) yes
checking for string.h... (cached) yes
checking for unistd.h... (cached) yes
checking for stdbool.h that conforms to C99... yes
checking for _Bool... no
checking for an ANSI C-conforming const... yes
checking for inline... inline
checking for size_t... yes
checking whether struct tm is in sys/time.h or time.h... time.h
checking how to run the C preprocessor... gcc -E
checking if zlib is wanted... checking for inflateEnd in -lz... yes
checking zlib.h usability... yes
checking zlib.h presence... yes
checking for zlib.h... yes
```

```
checking for inflateEnd in -lz... (cached) yes
checking zlib in /usr... ok
checking for boostlib >= 1.20.0... configure: error: We could not
detect the boost libraries (version 1.20 or higher). If you have
a staged boost library (still not installed) please specify $BOOST_ROOT
in your environment and do not give a PATH to --with-boost option.
If you are sure you have boost installed, then check your version number
looking in <boost/version.hpp>. See http://randspringer.de/boost for more
documentation.
```

- Fíjese en el bloque en negrita: falta una librería. Esto se trata de un error, es pues muy importante instalar la librería en cuestión. Si **configure** no es más que una alerta (warning), tiene dos opciones:
  - instalar la librería que falta (y su paquete de desarrollo) bien desde las fuentes, bien desde el gestor de paquetes de su distribución;
  - no instalarlo: no es vital. Sin embargo, no habrá soporte de las fuentes Type1, lo que es un problema para los PDF.
- Después de haber resuelto, si lo desea, las dependencias, debe volver a ejecutar el comando **./configure**. El comando **configure** crea el archivo **Makefile** correspondiente. Este archivo contiene el conjunto de las reglas, rutas y opciones para compilar el programa. El comando **make** depende de él.
- Inicie la compilación con el comando **make**. Puede que aparezcan alertas (líneas warning): no significa obligatoriamente que el programa no compilará o no funcionará posteriormente. De todas maneras, si la compilación produce errores, se parará por sí misma con un mensaje de error del compilador que puede a veces (pero no siempre) ponerle sobre la pista de una solución.

  La compilación puede ser más o menos larga según el producto compilado. Para PDFEdit, una máquina Intel Core 2 Duo a 3200 MHz requirió unos 6 minutos.

```
root@debian:/home/seb/pdfedit/pdfedit-0.4.5# make
cd /home/seb/pdfedit/pdfedit-0.4.5/src && make
make[1]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/xpdf && make staticlib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf'
cd goo && make
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/goo'
make[3]: No se hace nada para `all'.
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/goo'
cd fofi && make
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/fofi'
make[3]: No se hace nada para `all'.
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/fofi'
cd splash && make
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/splash'
make[3]: No se hace nada para `all'.
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/splash'
cd xpdf && make staticlib
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/xpdf'
make[3]: No se hace nada para `staticlib'.
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/xpdf'
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/utils && make staticlib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/utils'
make[2]: No se hace nada para `staticlib'.
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/utils'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/kernel && make staticlib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/kernel'
make[2]: No se hace nada para `staticlib'.
```

```
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/kernel'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/xpdf && make extralib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf'
cd xpdf && make extralib
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/xpdf'
make[3]: No se hace nada para `extralib'.
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf/xpdf'
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/xpdf'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/tests/ && make
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/tests'
make[2]: No se hace nada para `all'.
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/tests'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/kpdf-kde-3.3.2 && make -f
Makefile.gui staticlib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/kpdf-kde-3.3.2'
make[2]: No se hace nada para `staticlib'.
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/kpdf-kde-3.3.2'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/qsa && make staticlib
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa'
make -f Makefile.qsa
make[3]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa'
cd src && make -f Makefile
make[4]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa/src'
cd qsa && make -f Makefile
make[5]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa/src/qsa'
make[5]: No se hace nada para `first'.
make[5]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa/src/qsa'
make[4]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa/src'
make[3]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa'
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/qsa'
cd /home/seb/pdfedit/pdfedit-0.4.5/src/gui && make -f Makefile.gui
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/gui'
make[2]: No se hace nada para `first'.
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/gui'
make[1]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src'
```

- Al finalizar sin error la compilación, termine la instalación del producto con **make install**. Cuidado: el producto se va a instalar en /usr/local/, lo que requiere los derechos del usuario root. Según su distribución, deberá usar o bien **su**, o bien el comando **sudo**.

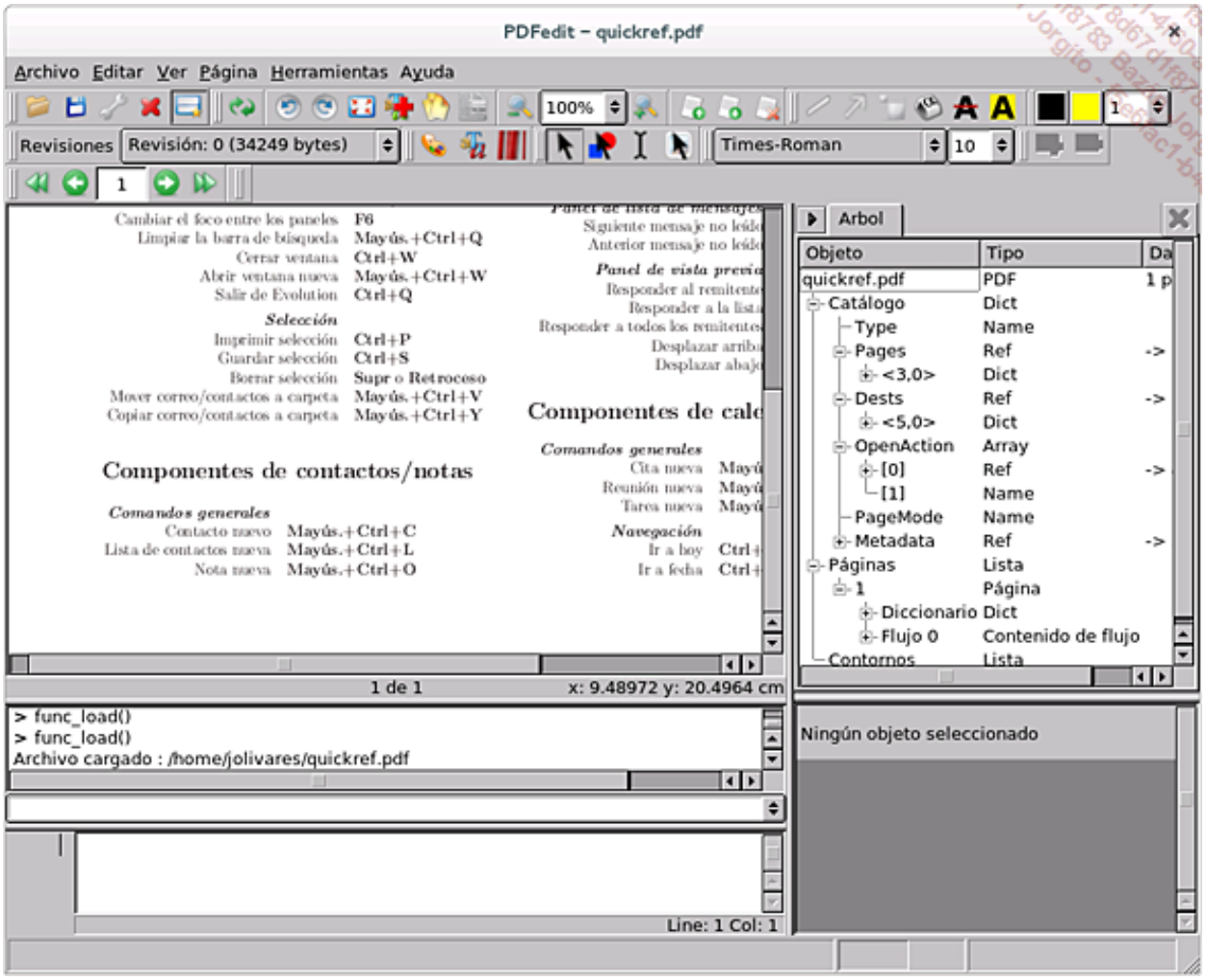
```
seb@debian:~/pdfedit/pdfedit-0.4.5$ su -c "make install"
...
cp -f "lang/pdfedit_es.qm" "/usr/local/share/pdfedit/lang/"
cp -f "lang/pdfedit_fr.qm" "/usr/local/share/pdfedit/lang/"
cp -f "lang/pdfedit_ru.qm" "/usr/local/share/pdfedit/lang/"
cp -f "lang/pdfedit_sk.qm" "/usr/local/share/pdfedit/lang/"
cp -f "lang/pdfedit_zh_TW.qm" "/usr/local/share/pdfedit/lang/"
cp -f "scripts/delinearize.qs" "/usr/local/share/pdfedit/scripts/"
cp -f "scripts/flatten.qs" "/usr/local/share/pdfedit/scripts/"
cp -f "scripts/pdftoxml.qs" "/usr/local/share/pdfedit/scripts/"
cp -f "scripts/update.qs" "/usr/local/share/pdfedit/scripts/"
cp -f "pdfedit" "/usr/local/bin/pdfedit"
make[1]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/src/gui'
cd /home/seb/pdfedit/pdfedit-0.4.5/doc && make doc_dist
make[1]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc'
cd user && make pdfedit.1
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc/user'
make[2]: `pdfedit.1' está actualizado.
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc/user'
make[1]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc'
cd /home/seb/pdfedit/pdfedit-0.4.5/doc && make doc_dist_install
make[1]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc'
cd user && make pdfedit.1
make[2]: se ingresa al directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc/user'
make[2]: `pdfedit.1' está actualizado.
```

```
make[2]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc/user'
mkdir -p /usr/local/share/man/man1;
cp -f user/pdfedit.1 /usr/local/share/man/man1
mkdir -p /usr/local/share/doc/pdfedit/;
cp -f LICENSE.GPL ../README ../Changelog AUTHORS README-* \
                README.cygwin /usr/local/share/doc/pdfedit
make[1]: se sale del directorio `/home/seb/pdfedit/pdfedit-0.4.5/doc'
seb@debian:~/pdfedit/pdfedit-0.4.5$
```

Inicie el producto:

```
$ pdfedit
```

Felicidades, la compilación y la instalación han funcionado perfectamente.



PDFEdit, compilado e instalado desde las fuentes

4. Desinstalación

La mayoría de los **Makefile**, al menos los generados por **configure**, permiten la desinstalación. Se efectúa con el comando **make uninstall**.

```
$ su -c "make uninstall"
password:
...
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_commandline.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_editor.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_execute.png"
```

```
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_lookandfeel.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_objecttree.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_paths.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/options_toolbars.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/pagespace.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/pdfedit.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_all.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_catalog.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_edit.png"
rm -f -r "/usr/local/share/doc/pdfedit/gui/images/propedit_edit_add.png"
...
```

5. Las bases del Makefile

a. Bases

El programa **make** utiliza un archivo Makefile para ejecutar un conjunto de acciones como la compilación de un proyecto, pero este archivo no se limita a eso: se trata de una especie de script por niveles.

Supongamos el proyecto siguiente, encargado de que aparezca "Hola". Muchas cosas para un ejemplo tan limitado, pero se trata de un ejemplo.

```
$ cat hola.h
#ifndef H_HOLA
#define H_HOLA
void Hola(void);
#endif

$ cat hola.c
#include <stdio.h>
#include <stdlib.h>

void Hola(void)
{
        printf("Hola\n");
}

$ cat main.c
#include <stdio.h>
#include <stdlib.h>
#include "hola.h"

int main(void)
{
        Hola();
        return 0;
}
```

Para compilar este proyecto a mano, debe ejecutar las etapas siguientes:

```
$ gcc -o hola.o -c hola.c
$ gcc -o hola hola.o main.o
$ gcc -o main.o -c main.c
$ ./hola
Hola
```

El Makefile se compone de reglas con la estructura siguiente:

```
destino: dependencia
comandos
```

Por lo tanto, un primer Makefile podría ser:

```
$ cat Makefile
hola: hola.o main.o
     gcc -o hola hola.o main.o

hola.o: hola.c
     gcc -o hola.o -c hola.c

main.o: main.c hola.h
     gcc -o main.o -c main.c
```

Primera regla: para ejecutar la regla hola, hace falta disponer de los archivos **`hola.o`** y **`main.o`**. Si se dispone de ellos, hay que ejecutar el comando **gcc -o hola hola.o main.o**.

Segunda regla: para ejecutar la regla `hola.o` hace falta disponer del archivo **`hola.c`**. Si está presente, entonces se ejecuta el comando **gcc -o hola.o -c hola.c**.

Tercera regla: para ejecutar la regla `main.o` hace falta disponer de los archivos **`main.c`** y**`hola.h`**. Si están presentes, entonces se ejecuta el comando **gcc -o main.o -c main.c**.

Las dos últimas reglas permiten resolver la primera. Si usted ejecuta el comando **make**, éste va a determinar cuáles son las reglas aplicables, en qué orden y luego va a aplicarlas en el orden de las dependencias. Si los archivos están actualizados, **make** no vuelve a construirlos a no ser que hayan sido modificados.

```
$ rm -f *.o
$ make
gcc -o hola.o -c hola.c
gcc -o main.o -c main.c
gcc -o hola hola.o main.o
$ ./hola
Hola
```

b. Makefile intermedio

El anterior Makefile funciona, pero no es óptimo:

- No permite compilar varios binarios.
- No permite limpiar los archivos temporales (.o) después de la compilación.
- No permite forzar la compilación del proyecto.

El añadido de nuevas reglas permite paliar estos problemas:

- all: genera n reglas;
- clean: limpia los .o;
- mrproper: llama a clean y suprime los binarios.

```
$ cat Makefile
all: hola

hola: hola.o main.o
        gcc -o hola hola.o main.o

hola.o: hola.c
        gcc -o hola.o -c hola.c

main.o: main.c hola.h
        gcc -o main.o -c main.c
```

```
clean:
        rm -rf *.o

mrproper: clean
        rm -rf hola

$ make clean
rm -rf *.o
$ make mrproper
rm -rf *.o
rm -rf hola
$ make all
gcc -o hola.o -c hola.c
gcc -o main.o -c main.c
gcc -o hola hola.o main.o
```

c. Un poco más complejo

Variables de usuario

Para terminar esta pequeña presentación, puede definir variables en su archivo y utilizar variables internas predefinidas.

El Makefile se convierte en:

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=hola

all: $(EXEC)

hola: hola.o main.o
        gcc -o hola hola.o main.o $(LDFLAGS)

hola.o: hola.c
        gcc -o hola.o -c hola.c $(CFLAGS)

main.o: main.c hola.h
        gcc -o main.o -c main.c $(CFLAGS)

clean:
        rm -rf *.o

mrproper: clean
        rm -rf $(EXEC)
```

Variables internas

Entre las variables internas:

- $@: nombre del destino.
- $<: nombre de la primera dependencia.
- $^: lista de las dependencias.
- $?: dependencias más recientes que la actual.
- $*: nombre del archivo sin el sufijo.

El Makefile se convierte en:

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=hola

all: $(EXEC)

hola: hola.o main.o
        gcc -o $@ $^ $(LDFLAGS)

hola.o: hola.c
        gcc -o $@ -c $ $(CFLAGS)

main.o: main.c hola.h
        gcc -o $@ -c $ $(CFLAGS)

clean:
        rm -rf *.o

mrproper: clean
          rm -rf $(EXEC)
```

Reglas de inferencia

Existen reglas predefinidas, basadas en accesos directos, que permiten generar metas en función del nombre del archivo C y objeto: **%.o: %.c**. La tentación de crear una regla única para **main.o** y **hola.o** es grande:

```
%.o: %.c
        gcc -o $@ -c $< $(CFLAGS)
```

Esta regla es correcta, pero falta la dependencia del header **hola.h**: si se modifica éste, ya no se compila el proyecto. Hay que añadir una regla específica:

```
main.o: hola.h
```

El Makefile se convierte en:

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=hola

all: $(EXEC)

hola: hola.o main.o
        gcc -o $@ $^ $(LDFLAGS)

%.o: %.c
        gcc -o $@ -c $< $(CFLAGS)

main.o: hola.h

clean:
        rm -rf *.o

mrproper: clean
       rm -rf $(EXEC)
```

Gestionar las librerías compartidas

1. Fundamentos

Una librería compartida es un archivo particular que contiene una lista de funciones, o API, accesible a cualquier programa que lo necesite sin tener que volver a escribirlas. Al contrario de lo que sucede con las librerías estáticas, el programa accede de manera dinámica a las funciones situadas en un archivo aparte. N programas diferentes pueden acceder a las funciones propuestas por la librería. Las librerías agrupan funciones propias de un dominio o un conjunto de dominios coherentes: tratamiento de imágenes, sonido, acceso a una base de datos, etc.

Un conjunto de funciones propuestas por una o varias librerías compartidas forma una **API**,*Application Programming Interface*, y a veces se agrupan en un framework que ofrece una solución completa para un dominio dado.

Se establece un vínculo entre el programa y una librería compartida durante la etapa de la edición de los vínculos por el editor de vínculos **ld**, al cual puede llamar el compilador **gcc** con la opción **-l<lib>**.

Otra posibilidad para un programa consiste en utilizar la función **C dlopen**, que abre una librería dinámica como un archivo y que accede a las funciones contenidas mediante punteros de funciones.

Si un programa depende de una librería compartida y ésta está ausente, el programa no podrá funcionar nunca más.

En Linux (y Unix en general), las librerías compartidas se llaman **Shared Objects** (so) en el sentido en el que se trata de archivos de objetos sin bloque de instrucción **main**. Los nombres de los archivos asociados llevan el sufijo **.so**.

Una librería puede disponer de varias versiones, que pueden ser o no compatibles, y se puede precisar la versión durante la edición de los vínculos. También es posible determinar una versión por defecto.

2. Lugar de almacenamiento

Por convención se colocan las librerías compartidas en directorios llamados lib:

- **/lib**: librerías de sistema básicas, vitales.
- **/lib64**: librerías de sistema de 64 bits.
- **/usr/lib**: librerías de usuario básicas, no necesarias para el boot.
- **/usr/local/lib**: librerías locales para los programas para la máquina.
- **/usr/X11R6/lib**: librerías del entorno X Window.
- ...

```
$ ls -l /lib
total 6024
...
-rwxr-xr-x  1 root root  114636 oct 23  2007 ld-2.6.1.so
lrwxrwxrwx  1 root root      11 oct  5  2007 ld-linux.so.2 -> ld-2.6.1.so
lrwxrwxrwx  1 root root      13 oct  5  2007 ld-lsb.so.2 -> ld-linux.so.2
lrwxrwxrwx  1 root root      13 oct  5  2007 ld-lsb.so.3 -> ld-linux.so.2
lrwxrwxrwx  1 root root      15 oct  5  2007 libacl.so.1 -> libacl.so.1.1.0
-rwxr-xr-x  1 root root   27864 sep 22  2007 libacl.so.1.1.0
lrwxrwxrwx  1 root root      15 oct  5  2007 libaio.so.1 -> libaio.so.1.0.1
-rwxr-xr-x  1 root root    5248 sep 21  2007 libaio.so.1.0.1
-rwxr-xr-x  1 root root   10256 oct 23  2007 libanl-2.6.1.so
lrwxrwxrwx  1 root root      15 oct  5  2007 libanl.so.1 -> libanl-2.6.1.so
```

```
lrwxrwxrwx  1 root root      20 oct  5  2007 libapparmor.so.1 ->
libapparmor.so.1.0.2
-rwxr-xr-x  1 root root   30404 sep 22  2007 libapparmor.so.1.0.2
lrwxrwxrwx  1 root root      16 oct  5  2007 libattr.so.1 ->
libattr.so.1.1.0
-rw-r--r--  1 root root   18272 sep 21  2007 libattr.so.1.1.0
...
```

En las distribuciones Linux recientes (las basadas en systemd), los directorios /lib y /lib64 ya no están en la raíz, su contenido se ha cambiado a /usr/lib y /usr/lib64.

La librería más importante del sistema es la C. Todos los programas compilados están vinculados a libc. Basta suprimir este archivo (un error de principiante) para hacer caer todo el sistema.

```
$ ls -l libc.so.6
lrwxrwxrwx 1 root root 13 oct 5 2007 libc.so.6 -> libc-2.6.1.so
```

Los directorios de las librerías contienen muchos vínculos simbólicos. Estos vínculos están aquí, entre otras cosas, para gestionar las versiones y la compatibilidad entre las versiones (como sucede, por ejemplo, cuando dos versiones coexisten).

```
$ cd /usr/lib
$ ls -l libXm.*
lrwxrwxrwx 1 root root      14 oct 17  2007 libXm.so.3 -> libXm.so.3.0.3
-rwxr-xr-x 1 root root 2371164 oct 12  2007 libXm.so.3.0.3
lrwxrwxrwx 1 root root      14 oct  5  2007 libXm.so.4 -> libXm.so.4.0.0
-rwxr-xr-x 1 root root 2496528 sep 22  2007 libXm.so.4.0.0
```

3. ¿Qué librerías vinculadas hay?

El comando **ldd** permite determinar cuáles son las librerías vinculadas a un programa, y también si están presentes o no.

```
$ ldd pdfedit
        linux-gate.so.1 =>  (0xffffe000)
        libfreetype.so.6 => /usr/lib/libfreetype.so.6 (0xb7ea5000)
        libz.so.1 => /lib/libz.so.1 (0xb7e92000)
        libt1.so.5 => /usr/lib/libt1.so.5 (0xb7e3d000)
        libqt-mt.so.3 => /usr/lib/libqt-mt.so.3 (0xb7739000)
        libXext.so.6 => /usr/lib/libXext.so.6 (0xb772a000)
        libX11.so.6 => /usr/lib/libX11.so.6 (0xb760f000)
        libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0xb7520000)
        libm.so.6 => /lib/libm.so.6 (0xb74fb000)
        libgcc_s.so.1 => /lib/libgcc_s.so.1 (0xb74ef000)
        libc.so.6 => /lib/libc.so.6 (0xb73bc000)
        libpthread.so.0 => /lib/libpthread.so.0 (0xb73a5000)
        libpng12.so.0 => /usr/lib/libpng12.so.0 (0xb7380000)
        libXi.so.6 => /usr/lib/libXi.so.6 (0xb7376000)
        libXrender.so.1 => /usr/lib/libXrender.so.1 (0xb736d000)
        libXrandr.so.2 => /usr/lib/libXrandr.so.2 (0xb7366000)
        libXcursor.so.1 => /usr/lib/libXcursor.so.1 (0xb735c000)
        libXinerama.so.1 => /usr/lib/libXinerama.so.1 (0xb7358000)
        libXft.so.2 => /usr/lib/libXft.so.2 (0xb7345000)
        libfontconfig.so.1 => /usr/lib/libfontconfig.so.1 (0xb7318000)
        libSM.so.6 => /usr/lib/libSM.so.6 (0xb730f000)
        libICE.so.6 => /usr/lib/libICE.so.6 (0xb72f6000)
        libdl.so.2 => /lib/libdl.so.2 (0xb72f2000)
        libXau.so.6 => /usr/lib/libXau.so.6 (0xb72ee000)
        libxcb-xlib.so.0 => /usr/lib/libxcb-xlib.so.0 (0xb72ea000)
        libxcb.so.1 => /usr/lib/libxcb.so.1 (0xb72d1000)
        /lib/ld-linux.so.2 (0xb7f3c000)
        libXfixes.so.3 => /usr/lib/libXfixes.so.3 (0xb72cb000)
```

```
        libexpat.so.1 => /lib/libexpat.so.1 (0xb72aa000)
```

Pongamos ahora por caso que falta una librería (movida voluntariamente para poder hacer esta demostración):

```
$ ldd /usr/bin/esd
        linux-gate.so.1 =>  (0xffffe000)
        libwrap.so.0 => /lib/libwrap.so.0 (0xb7f6d000)
        libesd.so.0 => not found
        libasound.so.2 => /usr/lib/libasound.so.2 (0xb7eb1000)
        libaudiofile.so.0 => /usr/lib/libaudiofile.so.0 (0xb7e8e000)
        libm.so.6 => /lib/libm.so.6 (0xb7e69000)
        libc.so.6 => /lib/libc.so.6 (0xb7d36000)
        libdl.so.2 => /lib/libdl.so.2 (0xb7d31000)
        libpthread.so.0 => /lib/libpthread.so.0 (0xb7d1a000)
        /lib/ld-linux.so.2 (0xb7f9f000)
```

Falta la librería `libesd.so`.0. Es imposible ejecutar el programa:

```
$ ./esd
./esd: error while loading shared libraries: libesd.so.0: cannot
open shared object file: No such file or directory
```

4. Configurar la caché del editor de vínculos

La edición de los vínculos con una librería compartida es dinámica y el sistema lo hace en el momento de la ejecución del programa con la ayuda de la librería **ld.so**. El binario proporciona el nombre de las librerías a vincular con la ejecución, pero no la ruta. Las funciones de **ld.so**determinan la librería a utilizar en función de su nombre, entre las rutas que conocen.

Cualquier programa se vincula a la librería **ld.so** o más bien **ld-linux.so** (ld-linux.so.2).

El cargador de vínculos **ld.so** busca las librerías en varios lugares, entre los cuales, se encuentran, por este orden:

- las rutas precisadas en la variable de entorno **LD_LIBRARY_PATH**. Se separan las rutas, como para PATH, por ":".
- el contenido del archivo **/etc/ld.so.cache**, que contiene una lista compilada (formato binario) de las librerías encontradas en las rutas predefinidas.
- los directorios **/lib** y **/usr/lib**.

La búsqueda en **/lib** y **/usr/lib** está implícita. Por otro lado, el hecho de rellenar la variable**LD_LIBRARY_PATH** no impide la búsqueda de las librerías en otros sitios, si no está en una de las rutas de la lista.

Para evitar la instalación de una variable cuyo contenido se puede manejar con dificultad, `Id.so`propone una caché que puede modificar por sí mismo. Se construye la caché desde el contenido del archivo **/etc/ld.so.conf** y del comando **ldconfig**.

Este archivo contiene la lista de los directorios que contienen las librerías compartidas:

```
# cat /etc/ld.so.conf
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/lib/Xaw3d
/usr/i386-suse-linux/lib
/usr/local/lib
/opt/kde3/lib
include /etc/ld.so.conf.d/*.conf
```

En vez de modificar este archivo, un paquete o usted mismo puede decidir añadir un archivo en **`/etc/ld.so.conf.d`** que contenga la ruta o las rutas de sus nuevas librerías.

No basta con añadir la ruta: debe regenerar la caché con el comando **ldconfig**.

```
# ldconfig
```

El comando **ldconfig**:

- actualiza la caché para las rutas definidas en /etc/ld.so.conf y asociadas, así como para /usr/lib y /lib;
- actualiza los vínculos simbólicos en las librerías;
- permite también listar las librerías conocidas en la caché.

Se aceptan las siguientes opciones:

Opción	Papel
-v	Modo verboso: indica lo que ldconfig efectúa
-N	No vuelve a construir la caché
-X	No actualiza los vínculos
-p	Lista el contenido de la caché

Para listar las librerías conocidas del editor de vínculos utilice:

```
# ldconfig -p
1940 libs encontradas en la caché «/etc/ld.so.cache»
        libzypp.so.324 (libc6) => /usr/lib/libzypp.so.324
        libzvbi.so.0 (libc6) => /usr/lib/libzvbi.so.0
        libzvbi-chains.so.0 (libc6) => /usr/lib/libzvbi-chains.so.0
        libzip.so.1 (libc6) => /usr/lib/libzip.so.1
        libzio.so.0 (libc6) => /usr/lib/libzio.so.0
        libz.so.1 (libc6) => /lib/libz.so.1
        libz.so (libc6) => /usr/lib/libz.so
        liby2util.so.3 (libc6) => /usr/lib/liby2util.so.3
        liby2storage.so.2 (libc6) => /usr/lib/liby2storage.so.2
        liby2.so.2 (libc6) => /usr/lib/liby2.so.2
        libycpvalues.so.3 (libc6) => /usr/lib/libycpvalues.so.3
        libycp.so.3 (libc6) => /usr/lib/libycp.so.3
        libx264gtk.so.54 (libc6) => /usr/lib/libx264gtk.so.54
        libx264gtk.so (libc6) => /usr/lib/libx264gtk.so
...
```

Para saber lo que haría **Idconfig** pero sin actualizar nada, use:

```
# ldconfig -N -X -v
/usr/X11R6/lib:
        libfglrx_pp.so.1.0 -> libfglrx_pp.so.1.0
        libfglrx_gamma.so.1 -> libfglrx_gamma.so.1.0
        libfglrx_tvout.so.1 -> libfglrx_tvout.so.1.0
        libGL.so.1 -> libGL.so.1.2
/usr/local/lib:
/opt/kde3/lib:
        libkdeinit_ksmserver.so -> libkdeinit_ksmserver.so
        libkdeinit_klipper.so -> libkdeinit_klipper.so
        libkdeinit_kcminit.so -> libkdeinit_kcminit.so
        libkdetvvideo.so.0 -> libkdetvvideo.so.0.0.0
        libkmailprivate.so -> libkmailprivate.so
...
```

Finalmente, para actualizar y ver el resultado use:

```
# ldconfig -v
```

(la salida es la misma que la del comando anterior).

El shell bash

1. Función

Aunque las distribuciones recientes de Linux permiten saltarse la introducción de instrucciones de texto al ofrecer entornos gráficos atractivos, un profesional de Linux no puede obviar el funcionamiento del intérprete de comandos y de los principales comandos asociados.

El intérprete de comandos, o simplemente intérprete, ejecuta las instrucciones introducidas con el teclado o en un script y le devuelve los resultados. Este intérprete es un programa comúnmente llamado shell. Se puede aproximar a la palabra kernel que vimos antes: el kernel significa núcleo. A menudo, está rodeado de una concha dura (piense en un hueso de albaricoque o melocotón). Como "shell" significa concha, viene a decir que es lo que rodea al **núcleo** de Linux: se utiliza mediante comandos. Por lo tanto, es una interfaz que funciona en modo texto entre el núcleo de Linux y los usuarios (avanzados) o las aplicaciones.

Hay varios shells: cada uno dispone de especificaciones propias. El Bourne Shell (sh) es el shell más conocido y habitual en los Unix. El C-Shell (csh) retoma la estructura del lenguaje C. El Korn Shell (ksh) es una evolución del Bourne Shell. El Z-Shell (zsh) es a su vez una evolución del Korn Shell. El shell de referencia en Linux se llama Bourne Again Shell (bash). A continuación le presentamos una lista exhaustiva de intérpretes de comandos que puede encontrar en Linux:

- sh: Thompson Shell (ya no existe);
- sh: Bourne Shell (sustituyó al anterior);
- bash: Bourne Again Shell;
- ksh: Korn Shell;
- csh: C Shell;
- zsh: Z Shell;
- tcsh: Tenex C Shell;
- ash: A Shell;
- dash: Debian Almquist Shell.

La lista de los shells disponibles en su instalación de Linux está en el archivo /etc/shells.

2. Bash: el shell por defecto

a. Un shell potente y libre

El bash es un derivado de Bourne Shell. Bourne es el nombre del principal programador de este shell. La expresión "Bourne Again" es un guiño a los orígenes del bash (Bourne) y un juego de palabras en "I born again", lo que significa "he nacido otra vez" o "reencarnado". El bash retoma sh pero también funcionalidades de ksh o chs.

El bash no sólo está en Linux. Al ser un programa libre, se puede compilar o ejecutar en numerosas plataformas. Es el shell de referencia en los sistemas Mac OS X y existe también para Windows.

El shell funciona en un terminal. Originalmente, un terminal es una verdadera máquina que sólo dispone de lo necesario para introducir instrucciones (el teclado) y visualizar los resultados (una pantalla o incluso hace mucho tiempo una simple impresora con papel listado). Aunque en la actualidad aún existan terminales físicos, en máquinas corrientes han sido sustituidos por programas que emulan terminales. Se distinguen dos tipos de ellos en Linux:

- las consolas virtuales de texto, el modo por defecto de Linux cuando arranca o funciona bajo entorno gráfico;
- las consolas o terminales gráficos, como xterm, eterm o konsole, que son emuladores de terminales en el seno de ventanas gráficas.

El shell funciona en un terminal. Espera entradas por teclado en la consola o la ventana, y visualiza sus resultados en el mismo lugar. Cualquier usuario avanzado de Linux o Unix en general tiene al menos un terminal abierto permanentemente. La apertura de un terminal (o consola, en este caso estas palabras son sinónimas) ejecuta automáticamente el shell por defecto.

b. Línea de comandos

El shell espera entradas por el teclado en una línea llamada línea de comandos o prompt. Un cursor, representado por un rectángulo fijo, intermitente o un carácter subrayado, indica la posición actual de su entrada.

La línea (prompt) proporciona información en el terminal y su posición en el sistema de archivos.

```
seb@slyserver:/home/public>
```

o

```
seb@slyserver:/home/public$
```

En esta clásica línea, obtiene cuatro datos:

- seb: es el nombre de inicio de sesión o login del usuario actualmente conectado al terminal;
- slyserver: es el nombre de anfitrión (hostname), el nombre lógico de la máquina conectada al terminal;
- /home/public: es la posición actual del shell en el sistema de archivos;
- > o $: es la terminación estándar del bash para un usuario sin privilegios.

Esta línea le informa de que el usuario sin privilegios de administración seb es el que utiliza el terminal (está conectado) en la máquina slyserver y que se encuentra actualmente en /home/public.

El carácter de terminación puede tener otros significados:

- $ indica que el usuario no tiene privilegios particulares, como con >.
- # indica que el usuario es el administrador root que tiene todos los privilegios.

La ruta puede variar:

seb@slyserver:~$: el carácter tilde ~ indica que se encuentra en su directorio personal.

seb@slyserver:~/test$: al combinarse con ~ (su directorio personal), indica que está en el subdirectorio test de éste.

La línea de comando se puede personalizar modificando una variable de entorno llamada PS1.

Para cualquier otra ruta se suele sustituir la línea de comandos por un simple símbolo de dólar, $, con el fin de ganar espacio.

3. Utilizar el shell

a. La introducción de datos

En el terminal, el teclado se utiliza de forma intuitiva. Puede desplazarse en la línea con las flechas de derecha e izquierda del teclado y borrar caracteres con las teclas [Tab] y [Supr]. Ejecute el

comando que ha introducido presionando la tecla [Entrar].

Hay varios métodos abreviados que son prácticos:

- **[Ctrl] A**: ir al principio de la línea.
- **[Ctrl] E**: ir al final de la línea.
- **[Ctrl] L**: borrar el contenido del terminal, y mostrar la línea de comandos en la parte superior.
- **[Ctrl] U**: borrar la línea hasta el principio.
- **[Ctrl] K**: borrar la línea hasta el final.

Ha llegado la hora de probar comandos. El comando **date** indica la fecha y la hora actuales. Por supuesto no obtendrá el mismo resultado, y no siempre en el mismo idioma; depende de su instalación de Linux.

```
$ date
mié nov 19 18:46:37 CET 2014
```

Un comando práctico, **pwd**, permite saber en qué directorio está.

```
$ pwd
/home/jolivares
```

El shell indica que está en el directorio /home/jolivares.

b. Sintaxis general de los comandos

Los comandos o instrucciones (las dos palabras son sinónimas en este caso) GNU tienen a menudo una sintaxis con la misma estructura:

```
Comando [parámetros] [argumentos]
```

Un comando puede no tener ni parámetros ni argumentos. En este caso, ejecuta la acción por defecto para la cual se programó, o muestra un mensaje de error, de haberlo.

Un parámetro es una opción del comando. Las dos palabras son sinónimas aquí. A menudo es una simple letra o una simple cifra precedida de un guión: `-l`, `-p`, `-s`, etc. Si el comando acepta varios parámetros, los puede introducir unos tras otros separándolos por espacio: `-l -r -t`, o escribiendo un solo guión y luego todos los parámetros: `-lrt`. Se aceptan las dos sintaxis: producen el mismo resultado. Simplemente, la segunda es más corta.

En ciertos casos, un parámetro necesita un argumento, por ejemplo un nombre de archivo. En este caso, es preferible separar este parámetro de los demás: `-lrt -f miarchivo` o de ubicarlo al final de los parámetros : `-lrtf miarchivo`.

Los argumentos son los valores en los que un comando debe ejecutar su acción. El tipo de valor (archivo, texto, números) depende del comando.

c. Primer ejemplo concreto con cal

Tomemos el ejemplo del comando **cal**. Admite varios parámetros y argumentos. Si se le invoca sin argumentos, muestra el calendario del mes en curso y resalta el día actual.

```
$ cal
   Noviembre 2014
```

```
do lu ma mi ju vi sá
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

El comando **cal** admite dos argumentos opcionales. Si se precisa sólo uno, se trata del año, y se muestra el calendario de ese año en su totalidad. Si se le indican dos argumentos, el primero es el mes; el segundo, el año.

```
$ cal 12 1975
    diciembre 1975
do lu ma mi ju vi sá
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

El comando admite también unos parámetros. Observará que por defecto se prevé la visualización para los anglosajones: la primera columna es un domingo, que representa el primer día de la semana. En España, es el lunes. El parámetro `-m` (de monday, en inglés) permite precisarlo:

```
$ cal -m 12 1975
    diciembre 1975
lu ma mi ju vi sá do
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Un segundo parámetro `-3` permite visualizar los meses que preceden y siguen al mes determinado (o el mes en curso).

```
jolivares@slyserver:~> cal -m -3 12 1975
   noviembre 1975         diciembre 1975          enero 1976
lu ma mi ju vi sá do  lu ma mi ju vi sá do  lu ma mi ju vi sá do
                1  2   1  2  3  4  5  6  7            1  2  3  4
 3  4  5  6  7  8  9   8  9 10 11 12 13 14   5  6  7  8  9 10 11
10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
24 25 26 27 28 29 30  29 30 31              26 27 28 29 30 31
```

Y como puede agrupar los parámetros, el comando siguiente produce el mismo resultado.

```
$ cal -m3 12 1975
```

d. Encadenar los comandos

Puede ejecutar varios comandos en una sola línea, unos tras otros. Para ello, basta con separarlos con un punto y coma.

```
$ date;pwd;cal
lun feb 25 22:29:09 CET 2008
/usr/share/man/man9
    febrero 2008
do lu ma mi ju vi sá
```

```
               1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

e. Visualizar texto con echo

No hay nada más sencillo que visualizar texto. El comando **echo** está hecho para ello. Como casi todos los comandos, acepta los parámetros, además de los argumentos, en forma de texto. Para visualizar un texto sencillo:

```
$ echo Hola amigos
Hola amigos
```

Puede colocar el texto entre comillas o simples apóstrofes para aclarar y agrupar el texto que va a visualizar. Podrá ver más adelante que esto tiene un significado particular.

Observe que, por defecto, su texto se visualiza y echo efectúa solo un retorno de carro para empezar otra línea. Puede modificar su texto para añadirle secuencias de caracteres que tienen una acción particular. Si ya conoce el lenguaje C, son las mismas. En la lista se encuentran sólo las más utilizadas:

Secuencia	Acción
\n	Pasar a la línea
\t	Tabulación horizontal
\c	Suprimir el salto de línea final
\b	Retorno de un carácter atrás
\\	Visualiza la barra oblícua
\nnn	Visualiza el carácter especificado en octal

Para utilizar estas secuencias, añada el argumento `-e`:

```
$ echo -e "Hola.\tMe llamo Javier\b\b\bNadie\n"
Hola. Me llamo Nadie
```

f. Comandos internos y externos

Existen dos tipos de comandos:

- Los comandos externos son programas binarios presentes como archivos en su disco duro (o cualquier otro soporte de datos). Cuando ejecuta el comando, se carga este archivo en memoria y se inicia como proceso (esta noción se explicará en este mismo capítulo).
- Los comandos internos son del proprio shell y se ejecutan en él. Estos comandos forman parte del programa shell, el bash. Los comandos **cd** o **pwd** son dos ejemplos de comandos internos. Cuando los ejecuta, el shell ejecuta las funciones definidas en su interior que les corresponden.

Puede distinguir un comando interno de un comando externo con la ayuda del comando interno**type**. Así, **date** es un comando externo. Puede observar que es un archivo presente en **/bin**, mientras que **pwd** es interno al shell.

```
$ type date
date is /bin/date
$ type pwd
pwd es una orden interna del shell
```

También puede encontrar otros tipos, como los alias de comandos que son atajos de comandos propios del shell. Así el shell bash de ciertas distribuciones Linux proponen alias como ll, que corresponde en realidad a ls -l.

```
$ type ll
ll is aliased to `ls -l'
```

g. Algunos atajos útiles

Se deben conocer algunas secuencias de atajos de comandos:

- **[Ctrl] C**: interrupción del programa: se termina.
- **[Ctrl] Z**: para el programa (ver los procesos).
- **[Ctrl] D**: interrumpe una introducción de datos en un símbolo del sistema >.

4. El historial de comandos

Le resultará muy útil poder volver a llamar a un comando que ya ejecutó navegando por el historial de los comandos con las teclas [Flecha arriba] y [Flecha abajo]. La flecha arriba vuelve atrás en el historial. Si ha tecleado los dos comandos anteriores (**date** y luego **pwd**), al presionar una primera vez en la flecha arriba se visualiza en la línea de comandos **pwd**, y a la segunda se visualiza el comando **date**. La flecha abajo navega en el otro sentido hasta la línea de origen. Si pulsa la tecla [Entrar] vuelve a ejecutar el comando.

Cuantos más comandos teclee, más se amplía el historial. El shell conserva así un gran número de entradas en el historial (se puede modificar el número de líneas guardadas). Este historial se conserva en un archivo caché de su directorio personal llamado .bash_history. Puede ver el contenido del historial con el comando **history**. El resultado siguiente está truncado de manera voluntaria, ya que la lista es demasiado larga.

```
$ history
...
 1000  date
 1001  pwd
 1002  uname -a
 1003  ls
 1004  fc -l -5
 1005  history
```

El comando **fc** funciona como history cuando se utiliza con el parámetro `-l`. Por defecto, se limita a los últimos quince comandos. También puede pasar el número de últimos comandos, como a continuación para los diez últimos:

```
$ fc -l -10
995      ssh -X seb@192.168.1.130
996      fc -l
997      fc -l -20
998      ls
999      pwd
1000     cd
1001     uname -a
1002     fc -l
1003     cat /etc/passwd
1004     ls -lrtR
```

Puede volver a llamar a un comando con fc y el parámetro `-s` seguido del número del comando. Entonces se ejecutará de manera automática.

```
$ fc -s 1001
uname -a
Linux slyserver 2.6.22.17-0.1-default #1 SMP 2008/02/10 20:01:04 UTC
x86_64 x86_64 x86_64 GNU/Linux
```

Finalmente, puede sustituir un elemento del comando por otro antes de ejecutarlo. Por ejemplo, puede sustituir `fc` por `ls` en la entrada 1002 del historial:

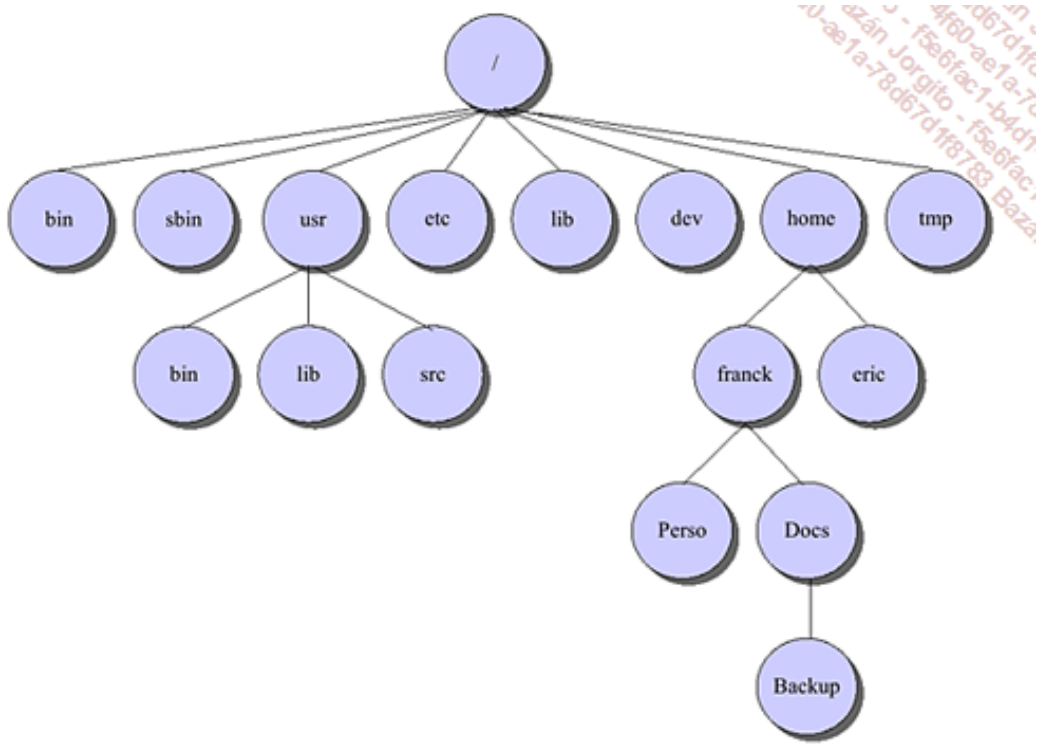
```
$ fc -s fc=ls 1002
ls -l
total 775944
-rw-r--r--  1 seb users     15391 may 14  2007 AR-1179161176460.pdf
...
```

La gestión de los archivos

1. El sistema de archivos

Un sistema de archivos, llamado comúnmente File System o FS, determina la organización de los datos en un soporte de almacenamiento, y por tanto, cómo gestiona y organiza el sistema operativo los archivos.

Linux es, como todo Unix, un sistema operativo completamente orientado a archivos. Se representa todo (o casi todo) con un archivo, tanto los datos (archivos de datos de cualquier tipo, como una imagen o un programa) como los periféricos (terminales, ratones, teclado, tarjeta sonido, etc.) o incluso los medios de comunicación (sockets, tuberías nombradas, etc.). Se puede decir que el sistema de archivos es el corazón de cualquier sistema Unix.



Ejemplo de árbol de directorio Linux

El sistema de archivos de Linux es jerárquico. Describe un árbol de directorios y subdirectorios, a partir de un elemento básico llamado raíz o root directory.

2. Los diferentes tipos de archivos

Distinguimos tres tipos de archivos: ordinarios, catálogo, especiales.

a. Los archivos ordinarios o regulares

Los archivos ordinarios se llaman también archivos regulares, ordinary files o regular files. Son archivos totalmente clásicos que contienen datos. Por datos se debe entender cualquier contenido:

- texto;
- imagen;
- audio;
- programa binario compilado;

- script;
- base de datos;
- librería de programación;
- etc.

Por defecto, nada permite diferenciar unos de otros, salvo la utilización de algunas opciones de determinados comandos (ls -F por ejemplo) o el comando **file**.

```
$ file nom_arch
nom arch: 32 Bits ELF Executable Binary (stripped)
```

Linux desconoce la noción de extensión de archivo como componente interno de la estructura del sistema de archivos. Dicho de otro modo, una extensión no es relevante dentro de un sistema de archivos y se la considera simplemente como parte del nombre. Sólo sirve para distinguir visual y rápidamente el posible contenido de un archivo en comparación con otro. No debemos hablar de extensión, sino de sufijo. Sin embargo, el primer término forma parte del lenguaje común, y se puede seguir utilizando: todo el mundo comprenderá lo que quiere decir.

Como Linux no gestiona las extensiones, el nombre de un programa no termina casi nunca por un ".exe"; habrá que encontrar otro método para distinguirlo.

b. Los catálogos

Los archivos catálogo son los directorios o carpetas. Los directorios permiten organizar el disco duro creando una jerarquía. Un directorio puede contener archivos normales, archivos especiales y otros directorios de manera recursiva.

Un directorio no es más que un archivo particular que contiene la lista de los propios archivos presentes en este directorio. Esta noción resultará muy útil cuando se trate el tema de los permisos.

c. Los archivos especiales

El tercer tipo de archivos es el especial. Existen varios tipos de archivos especiales. Por ejemplo, los drivers de los periféricos están representados por archivos especiales de la carpeta /dev.

Son principalmente archivos que sirven de interfaz para los diversos periféricos. Se pueden utilizar, según el caso, como archivos normales. Cuando se accede en modo lectura o escritura a estos archivos se redirigen hacia el periférico (pasando por el driver asociado si existe). Por ejemplo, si dirige un archivo de onda sonora (wave) hacia el archivo que representa la salida de la tarjeta de sonido, hay muchas probabilidades que este sonido sea audible por sus altavoces.

3. Nomenclatura de los archivos

No se puede dar cualquier nombre a un archivo; hay que seguir unas simples reglas, válidas para todos los tipos de archivos.

En los antiguos sistemas Unix, un nombre de archivo no podía superar los 14 caracteres. En los sistemas actuales, Linux incluido, se puede llegar hasta 255 caracteres. La posible extensión está incluida en la longitud del nombre del archivo.

Un punto extremamente importante: Linux respeta la distinción entre los nombres de archivos en minúsculas y en mayúsculas. Pepito, PEPITO, PePito y pepito son nombres de archivos diferentes, con un contenido diferente. Esta distinción es intrínseca al tipo de sistema de archivos. En otros sistemas de tipo Unix (como Mac OS X) este comportamiento puede ser opcional.

Se acepta la mayoría de los caracteres (las cifras, las letras, las mayúsculas, las minúsculas, ciertos signos, los caracteres acentuados), incluido el espacio. Sin embargo, se deben evitar algunos caracteres, ya que tienen un significado particular dentro del shell: & ; ( ) ~ <espacio> \ / | ` ? - (al principio del nombre).

Los nombres siguientes son válidos:

- Archivo1
- Paga.txt
- 123tratamiento.sh
- Paga_junio_2002.xls
- 8

Los nombres siguientes, aunque válidos, pueden crear problemas:

- Archivo*
- Pago(diciembre)
- Ben&Nuts
- Paga junio 2002.xls
- -f

4. Las rutas

a. Estructura y nombre de ruta

Las rutas permiten definir una ubicación en el sistema de archivos. Es la lista de los directorios y subdirectorios utilizados para acceder a un sitio determinado de la estructura hasta la posición deseada (directorio, archivo). Se suele completar un nombre de archivo con su ruta de acceso.

Por eso el archivo pepito del directorio dir1 es diferente del archivo pepito del directorio dir2. Al ser jerárquico, el sistema de archivos de Unix tiene forma de estructura en árbol.

El esquema presentado en la sección La gestión de los archivos - El sistema de archivos de este capítulo representa una estructura en árbol de un sistema de archivos Linux. La / situada arriba del todo se llama raíz o root directory (no confundir con el directorio del administrador root). El nombre de la ruta o path name de un archivo es la concatenación, desde la raíz, de todos los directorios que se deben cruzar para acceder a él, que están separados cada uno por el carácter /. Es una ruta absoluta, como la siguiente:

```
/home/pepito/Docs/Backup/fic.bak
```

Una ruta absoluta o completa:

- empieza desde la raíz. Por lo tanto, comienza con una /,
- describe todos los directorios que hay que cruzar para acceder al sitio deseado,
- no contiene ni . ni . . .

b. Directorio personal

Al crear un nuevo usuario, el administrador le asigna un directorio personal llamado home directory. Cuando inicia sesión, el usuario es dirigido directamente a ese directorio, que es el suyo personal, en el que podrá crear sus propios archivos y subdirectorios.

```
Login: seb
Password: xxxxxxxxxx
$ pwd
/home/seb
```

c. Ruta relativa

Un nombre de ruta también puede ser relativo a su posición en el directorio actual. El sistema (o el shell) recuerda la posición actual de un usuario en el sistema de archivos, el directorio activo. Puede acceder a otro directorio de la estructura desde su ubicación actual sin teclear la ruta completa, con sólo precisar la ruta más corta en relación con su posición actual dentro de la estructura.

Para ello, a menudo hace falta utilizar dos entradas particulares de directorios:

- El punto . representa el directorio corriente, activo. Suele estar implícito.
- Los dos puntos .. representan el directorio de nivel superior.

Una ruta relativa:

- describe una ruta relativa a una posición determinada en la estructura, en general (pero no siempre) desde la posición actual;
- describe en principio la ruta más corta para ir de un punto a otro;
- puede contener puntos o dos puntos.

Las tres afirmaciones anteriores no son obligatorias:

- `/usr/local/bin` es una ruta completa o absoluta;
- `Documents/Photos` es una ruta relativa: se considera que existe el directorio Documents en el directorio corriente;
- `./Documents/Photos` es una ruta relativa perfectamente idéntica a la anterior, con la salvedad de que el punto indica el directorio activo (corriente) de manera explícita. "./Documents" indica de manera explícita el directorio Documents en el directorio activo;
- `/usr/local/../bin` es una ruta relativa: los .. son relativos a /usr/local y suben un nivel hacia /usr. La ruta final es, por lo tanto /usr/bin.

d. La virgulilla

El bash interpreta el carácter virgulilla ~ como un alias del directorio personal. Las rutas pueden ser relativas a la virgulilla, pero ésta no debe ir precedida por carácter alguno. Para desplazarse en el directorio tmp de su carpeta personal esté donde esté:

```
$ cd ~/tmp
```

Si introduce esto, obtendrá un error:

```
$ cd /~
```

e. cd

Para desplazarse por los directorios, utilice el comando **cd** (*change directory*). El comando **pwd**(*print working directory*), que ya hemos comentado, muestra la ruta completa del directorio actual.

Si introduce **cd .**, no se mueve. El punto será muy útil cuando tenga que especificar rutas explícitas a comandos ubicados en el directorio donde está ubicado.

Cd .. sube un nivel. Si se encontraba en /home/seb, ahora estará en home.

El comando cd sin argumento permite volver directamente a su directorio de usuario.

A continuación, presentamos un ejemplo. El usuario seb se encuentra en su directorio personal. Se mueve mediante una ruta relativa hacia /home/public. Con .. sube hacia /home, por lo tanto con ../public se mueve a /home/public. De ahí, vía una ruta completa, se dirige hacia /urs/local/bon, y luego decide, con la ayuda de una ruta relativa, ir a /usr/lib: el primer .. baja hacia usr/local, el segundo hacia /usr, y luego vuelve hacia /usr/lib. Finalmente, seb vuelve a su directorio personal con cd sin argumento. Aquí se da la línea completa para una mejor comprensión.

```
seb@slyserver:~> pwd
/home/seb
seb@slyserver:~> cd ../public
seb@slyserver:/home/public> cd /usr/local/bin
seb@slyserver:/usr/local/bin> cd ../../lib
seb@slyserver:/usr/lib> cd
seb@slyserver:~>
```

5. Los comandos básicos

a. Listar los archivos y los directorios

El comando **ls** permite listar el contenido de un directorio (catálogo) en líneas o columnas. Soporta varios parámetros, de los cuales los más importantes son:

Parámetro	Significado
`-l`	Para cada archivo o carpeta, facilita información detallada.
`-a`	Se visualizan los archivos escondidos (empiezan por un punto).
`-d`	En un directorio, precisa el propio directorio, y no su contenido.
`-F`	Añade un carácter al final del nombre para especificar el tipo: / para un directorio, * para un ejecutable, @ para un vínculo simbólico, etc.
`-R`	Si el comando detecta subdirectorios, entra en ellos de manera recursiva.
`-t`	Se filtra la salida por fecha de modificación del más reciente al más antiguo. Se visualiza esta fecha.
`-c`	Muestra/ordena (con -t) por fecha de cambio de estado del archivo.
`-u`	Muestra/ordena (con -t) por fecha de acceso del archivo.
`-r`	Se invierte el orden de salida.
`-i`	Muestra el inodo del archivo.
`-C`	La visualización se hace en varias columnas (por defecto).
`-1`	La visualización se hace en una sola columna.

El parámetro que le facilita más información es `-l`: proporciona ciertos detalles relativos a los archivos.

```
$ ls -l
total 4568
-rw-r--r-- 1 seb users   69120 sep  3  2006 3i_recuperación_2006.doc
-rw-r--r-- 1 seb users    9632 sep  3  2006 3i_recuperación_2006.odt
-rw-r--r-- 1 seb users    6849 nov 17  2003 control_1I2_mártes.sxw
...
```

La línea de salida indica el tamaño total en bloques de 1024 bytes (o 512 bytes si se define una variable llamada POSIXLY_CORRECT) del contenido del directorio. Este tamaño representa el conjunto de los archivos ordinarios del directorio y no tiene en cuenta los posibles subdirectorios y su contenido (para ello, habrá que utilizar el comando **du**).

Luego viene la lista detallada de todo el contenido.

-rw-r--r--	1	seb	users	69120	sep 3 2006	3i_recuperación_2006.doc
1	2	3	4	5	6	7

1: El primer símbolo representa el tipo de archivo (-: ordinario, d: directorio, l: vínculo simbólico...); los otros, por bloques de tres, los permisos para el usuario (rw-), el grupo (r--) y todos (r--). Se explican los permisos en el capítulo Los discos y el sistema de archivos.

- 2: Un contador de vínculos (capítulo Los discos y el sistema de archivos).
- 3: El propietario del archivo, que suele ser su creador.
- 4: El grupo al cual pertenece el archivo.
- 5: El tamaño del archivo en bytes.
- 6: La fecha de la última modificación (a veces con la hora), siguiendo el parámetro (t, c, u).
- 7: El nombre del archivo.

Los permisos van a veces seguidos de un punto "." o de un signo más "+". El primero significa que el archivo dispone de un contexto de seguridad selinux, el segundo que el archivo dispone de permisos extendidos ACL. Estos conceptos no son abordados en este libro y no tienen ningún impacto en las operaciones propuestas en el libro.

Puede resultar muy útil la posibilidad de listar sus archivos de tal manera que se visualicen al final de la lista los modificados recientemente. Así, en caso de haber un gran número de archivos, tendrá delante estos últimos. El orden por fecha de modificación se hace con `-t`, y en el orden contrario, con `-r`. Añádale los detalles con `-l`.

```
$ ls -lrt
-rw-r--r-- 1 seb users   66107 ene  9 17:24 Parcial_1_1I_2008.pdf
-rw-r--r-- 1 seb users   13777 ene 10 17:58 parcial_3I_ppa_2007.odt
-rw-r--r-- 1 seb users   64095 ene 10 17:58 parcial_3I_ppa_2007.pdf
-rw-r--r-- 1 seb users  100092 feb 22 22:21 curso_shell_unix.odt
```

`ls -l -r -t` es estrictamente idéntico a `ls -lrt`, como ya se ha indicado en la sintaxis general de los comandos.

Un recurso mnemotécnico para recordar esta secuencia de argumentos es utilizarla bajo la forma `-rtl` (el orden de los argumentos no tiene importancia aquí) y pensar en la famosa radio europea RTL.

b. Gestionar los archivos y los directorios

Crear archivos vacíos

Quizás necesite crear archivos vacíos para hacer pruebas. Un comando práctico para ello es **touch**. Utilizado con el nombre de un archivo como argumento únicamente, crea un archivo con un tamaño cero.

```
$ touch fictest
$ ls -l fictest
-rw-r--r-- 1 seb users 0 feb 29 15:13 fictest
```

La creación de archivos vacíos no es el principal uso de touch. Si vuelve a ejecutar el mismo comando en el archivo, observará que la fecha de modificación ha cambiado. El manual de touch le informará de que así es posible modificar completamente la fecha y la hora de un archivo. Esto puede ser útil para forzar las copias de seguridad incrementales de archivos.

Crear directorios

El comando **mkdir** (*make directory*) permite crear uno o varios directorios, o una estructura completa. Por defecto, el comando no crea una estructura. Si pasa como argumentos dir1/dir2 y dir1 no existe, el comando devuelve un error. En este caso, utilice el parámetro `-p`.

```
mkdir [-p] dir1 [dir2] ... [dirn]
```

```
$ mkdir Documentos
$ mkdir Documentos/Fotos
$ mkdir -p Archivos/antiguallas
$ ls -R
.:
Archivos  Documentos  fictest

./Archivos:
antiguallas

./Archivos/antiguallas:

./Documentos:
Fotos
```

Suprimir directorios

El comando **rmdir** (*remove directory*) suprime uno o varios directorios. No puede suprimir una estructura. Si el directorio que pretende eliminar contiene archivos o directorios, el comando devuelve un error. Por lo tanto, el directorio no debe contener ni archivos ni directorios, y ello aunque los propios subdirectorios estén vacíos.

```
rmdir dir1 [dir2] ... [dirn]
```

No hay parámetro `-r` (recursividad) para el comando **rmdir**. Para suprimir una estructura tendrá que utilizar el comando **rm**.

```
$ rmdir Documentos/
rmdir: Documentos/: El directorio no está vacío.
$ rmdir Documentos/Fotos
$
```

Copiar archivos

El comando **cp** (copy) copia uno o varios archivos en otro archivo o en un directorio.

```
cp fic1 [fic2 ... ficn] Destino
```

En el primer caso, se vuelve a copiar fic1 en Destino. Si Destino existe, se sobreescribe sin aviso según el parámetro pasado y según los permisos. En el segundo caso, se copian de nuevo fic1, fic2

y así sucesivamente en el directorio Destino. Las rutas pueden ser absolutas o relativas. El comando admite, entre otras, las opciones siguientes:

Parámetro	Significado
`-i`	Pide confirmación de copia para cada archivo.
`-r`	Recursivo: copia un directorio y todo su contenido.
`-p`	Se preservan los permisos y fechas.
`-f`	Forzar la copia.
`-a`	Copia de archivo: el destino es en la medida de lo posible idéntico al origen. La copia es recursiva.

Preste atención al funcionamiento de cp con las copias de directorios. El funcionamiento es diferente según exista o no el directorio de destino. En el primer caso, dir2 no existe. Se copia el directorio dir1 en dir2. Al final dir2 es una copia exacta de dir1.

```
$ ls -d dir2
ls: no puede acceder a dir2: No existe el archivo o directorio
$ cp -r dir1 dir2
$ ls
dir1  dir2
```

Ahora que dir2 existe, ejecute de nuevo el comando cp. Esta vez, como dir2 existe, no será sobreescrito, como cabía esperar. El comando determina que, al ser dir2 el destino, se debe copiar dir1 en el destino: se copia dir1 en dir2.

```
$ cp -r dir1 dir2
$ ls dir2
dir1
```

Mover y volver a nombrar un archivo

El comando **mv** (*move*) permite mover, volver a nombrar un archivo o las dos cosas a la vez. Funciona como el comando cp. Los parámetros `-f` e `-i` tienen el mismo efecto. Con los tres comandos mv sucesivos siguientes:

- se vuelve a nombrar txt1 como txt1.old;
- se mueve txt2 a dir1;
- se mueve txt3 a dir1 y se vuelve a nombrar como txt3.old.

```
$ touch txt1 txt2 txt3
$ mv txt1 txt1.old
$ mv txt2 dir1/txt2
$ mv txt3 dir1/txt3.old
```

Observe la existencia del parámetro `-u`: si el archivo de destino es más reciente, impide que se sobreescriba.

Suprimir un archivo o una estructura

El comando **rm** (*remove*) suprime uno o varios archivos y, si es preciso, una estructura completa, según las opciones. La supresión es definitiva.

```
rm [Opciones] fic1 [fic2...]
```

Las opciones son las habituales, pero dada la particularidad y la peligrosidad del comando, parece

necesario repasarlas.

Parámetro	Significado
`-i`	El comando requerirá una confirmación para cada uno de los archivos que desea suprimir. Según la versión de Unix, el mensaje cambia y la respuesta también: y, Y, O, o, N, n, a veces todas.
`-r`	El parámetro siguiente que se espera es un directorio. En este caso, la supresión es recursiva: se suprimen todos los niveles inferiores, tanto los directorios como los archivos.
`-f`	Fuerza la supresión.

Por orden, los comandos siguientes suprimen un simple archivo, un directorio y una estructura de manera forzada:

```
$ rm fic1
$ rm -r dir1
$ rm -rf /home/public/depots
```

El uso combinado de los parámetros `-r` y `-f`, aunque muy útil y práctico, es muy peligroso, en particular como root. No se le pide ninguna confirmación. Al menos que se utilicen herramientas de recuperación de datos específicos, caras y poco eficaces, se perderán sus datos de manera irremediable. Existe un riesgo adicional: si cree que `rm -rf /` no tocará sus archivos con el pretexto de no contar con permisos en la raíz, ¡comete un error! El comando es recursivo, terminará llegando a su directorio personal…

Veamos un truquito. Suponga que dispone de un archivo cuyo nombre comienza con un guión. ¿Es posible suprimirlo con **rm**?

```
$ >-i # ver las redirecciones
$ rm -i
rm: missing operand
Pruebe: `rm --help' para más información.
```

Es imposible suprimir el archivo "-i" de esta manera, ya que **rm** lo interpreta como un parámetro, y no como un argumento. Por lo tanto hay que actuar con astucia. Existen dos soluciones:

- Utilizar la opción GNU `--` que significa el final de los parámetros y el principio de los argumentos.
- Añadir una ruta, relativa o completa, antes del guión.

Esta última solución tiene la ventaja de ser un estándar. Las dos líneas son equivalentes:

```
$ rm -- -i
$ rm ./-i
```

Los vínculos simbólicos

Puede crear vínculos simbólicos, que son un poco como atajos a un archivo. Un vínculo es un archivo especial que contiene como información la ruta hacia otro archivo. Es un tipo de alias. Existen dos tipos de vínculos: el vínculo duro (hard link), que veremos más adelante, durante el estudio de los sistemas de archivos, y el vínculo simbólico (soft link), que corresponde a la definición dada.

Es posible crear vínculos simbólicos hacia cualquier tipo de archivo, sea cual sea y esté donde esté. El comando de creación de vínculos simbólicos no comprueba la existencia del archivo al que se apunta. Es posible crear vínculos a archivos que no existen con el parámetro `-f`.

```
ln -s archivo vínculo
```

Si fuera necesario, el vínculo se comportará como el archivo al que se apunta, con los mismos permisos y las mismas propiedades:

- si el archivo al que se apunta es un programa, ejecutar el vínculo lleva a ejecutar el programa;
- si el archivo al que se apunta es un directorio, un cd sobre el vínculo entra en este directorio;
- si el archivo al que se apunta es un archivo especial (periférico), se ve el vínculo como periférico;
- etc.

Sólo en caso de que se elimine, el vínculo simbólico se "separa" del archivo al que apunta. La supresión de un vínculo simbólico implica la supresión de este vínculo únicamente, y no del archivo al que apunta. La supresión del archivo al que se apunta no implica la supresión de los vínculos simbólicos asociados. En este caso, el vínculo apunta al vacío.

```
$ touch fic1
$ ln -s fic1 vínculofic1
$ ls -l
-rw-r--r-- 1 seb users    0 mar  4 19:16 fic1
lrwxrwxrwx 1 seb users    4 mar  4 19:17 vínculofic1 -> fic1
$ ls -F
fic1  vínculofic1@
$ echo titi>fic1
$ cat vínculofic1
titi
```

Este ejemplo muestra que un vínculo simbólico es en realidad un archivo especial de tipo "l" que apunta hacia otro archivo. Observe en la lista detallada la presencia de una flecha que indica sobre qué archivo apunta el vínculo. Se distingue el carácter @ al indicar que se trata de un vínculo simbólico durante la utilización del parámetro `-F`. Si dispone de un terminal de color, es posible que el vínculo simbólico aparezca en azul claro (por convención en Linux). Si aparece en rojo, es que apunta al vacío.

Que un vínculo apunte al vacío no significa que no apunte a nada. Quizá esté hecho a propósito: es posible crear vínculos hacia puertos USB, o CD-ROM, entre otros, detrás de los cuales hay sistemas de archivos removibles. En este caso, el vínculo se vuelve activo cuando se inserta el soporte o se monta el sistema de archivos removido.

Se explicará más adelante el comando **echo** y el signo >. El efecto aquí es la escritura en el archivo fic1 de "titi". El comando **cat** visualiza el contenido de un archivo. Al representar el vínculo fic1, la salida es la esperada.

Cuidado, los permisos indicados son los del archivo especial y no tienen otro significado: no significa que todo el mundo tiene permisos en el archivo al que se apunta. Durante su utilización, son los permisos del archivo o de la carpeta a los que se apuntan los que prevalecen.

c. Comodines: carácter de sustitución

Al utilizar los comandos con el sistema de archivos, puede resultar interesante filtrar la salida de nombres de archivos con ayuda de determinados criterios, por ejemplo con el comando **ls**. En vez de visualizar toda la lista de archivos, se puede filtrar la visualización de varios criterios y caracteres especiales.

Caracteres(s)	Función
*	Sustituye una cadena de longitud variable, incluso vacía.
?	Sustituye cualquier carácter único.
[...]	Una serie o un rango de caracteres.
[a-b]	Un carácter entre el rango indicado (de a a b incluida).
[!...]	Inversión de la búsqueda.
[^...]	Ídem.

- Supongamos el contenido siguiente:

```
$ ls
afic   afic2  bfic   bfic2  cfic   cfic2  dfic   dfic2
afic1  afic3  bfic1  bfic3  cfic1  cfic3  dfic1  dfic3
```

- Obtiene todos los archivos que empiezan con a:

```
$ ls a*
afic1  afic2  afic3
```

- Todos los archivos de cuatro caracteres que empiezan con a:

```
$ ls a???
afic
```

- Todos los archivos de al menos tres caracteres y que empiezan con b:

```
$ ls b??*
bfic  bfic1  bfic2  bfic3
```

- Todos los archivos que terminan con 1 o 2:

```
$ ls *[12]
afic1  afic2  bfic1  bfic2  cfic1  cfic2  dfic1  dfic2
```

- Todos los archivos que empiezan con las letras de a a c, que tienen al menos un segundo carácter antes de la terminación 1 o 2:

```
$ ls [a-c]?*[12]
afic1  afic2  bfic1  bfic2  cfic1  cfic2
```

- Todos los archivos que no terminan por 3:

```
$ ls *[!3]
afic  afic1  afic2  bfic  bfic1  bfic2  cfic  cfic1  cfic2  dfic
dfic1  dfic2
```

Interpretación por el shell

El shell es el encargado de sustituir estos caracteres antes de pasar los parámetros a un comando. Así, en el momento de un `$ cp * Documents`, cp no recibe el carácter *, sino la lista de todos los archivos y directorios del directorio activo.

Los comodines pueden utilizarse dentro de todos los argumentos que representan archivos o rutas. Así, el comando siguiente va a volver a copiar todos los archivos README de todos los subdirectorios

de Documents en la posición actual:

```
$ cp Documents/*/README
```

d. Cierre de caracteres

Se deben cerrar algunos caracteres especiales; por ejemplo, en caso de caracteres poco corrientes en un nombre de archivo.

- La **contrabarra ** permite cerrar un carácter único. `ls paga\ *.xls` va a listar todos los archivos que contienen un espacio después de paga.
- Las **comillas** "..." permiten la interpretación de los caracteres especiales, de las variables, dentro de una cadena.
- Los **apóstrofes** '...' cierran todos los caracteres especiales en una cadena o un archivo.

Buscar archivos

1. Consideraciones generales

El comando **find** permite buscar archivos dentro de la estructura del sistema de archivos con la ayuda de criterios, y da la posibilidad de actuar sobre los resultados devueltos.

```
find ruta criterios opciones
```

Al ser recursivo el comando **find**, basta con indicar un directorio básico para que se desarrolle toda la estructura desde este directorio. La opción básica es `-print` (a menudo implícita en la mayoría de los Unix), que permite visualizar en pantalla los resultados.

```
seb@slyserver:~/Documents/slyunix> find
.
./logos-cuadrado.tif
./logos-cuadrado.eps
./Pagina 5.pdf
./logos-cuadrado-grande.jpg
./LOGOS
./sitio_2.jpg
./pub_planeta.pdf
./index_logon_inc.php
./logo-iceberg.eps
./flyer
./flyer/sly4.jpg
./flyer/flyerx4.sxd
./flyer/sly1.jpg
./flyer/sly2.jpg
./flyer/flyer.sxd
./flyer/sly3.jpg
./flyer/flyer.jpg
...
```

La visualización es relativa, ya que la ruta indicada es relativa. Si se hubiese especificado la ruta absoluta, la visualización habría sido absoluta.

2. Criterios de búsqueda

Los parámetros permiten definir los criterios de búsqueda. Si hay varios, puede combinarlos usando una Y lógica (criterio1 Y criterio2).

a. -name

`-name` permite una selección por nombres de archivos. Es posible utilizar los comodines ya vistos. Se coloca el criterio entre comillas. Aquí se visualiza la lista de todos los archivos desde la ubicación actual y que empiezan por "fic".

```
$ find . -name "fic*" -print
./fic1
./fic2
./fic3
./fic4
```

b. -type

-type permite una selección por tipo de archivos. Ya sabe que, además de los vínculos, los directorios y los archivos simples, están presentes otros tipos de archivos.

Código	Tipo de archivo
B	Archivo especial en modo bloque
C	Archivo especial en modo carácter
D	Directorio (directory)
F	Archivo ordinario
L	Vínculo simbólico
P	Tubería con nombre (pipe)
S	Socket (Conexión de red)

Se visualizan todos los directorios cuyo nombre empieza por "re".

```
$ find . -name "re*" -tipo d -print
./dir1
./dir2
```

c. -user y -group

-user y -group permiten una búsqueda sobre el propietario y el grupo de pertenencia de los archivos. Es posible precisar el nombre (usuario, grupo) o la ID (UID, GID). El ejemplo siguiente busca todos los archivos ordinarios que pertenecen a seb y al grupo users.

```
$ find . -tipo f -user seb -group users -print
./fic1
./fic3
```

d. -size

-size permite precisar el tamaño de los archivos buscados. Su sintaxis es particular, ya que trabaja por defecto en bloques si no se especifica nada. Puede resultar sorprendente a veces; más aún cuando el bloque, que aquí tiene un tamaño de 512 bytes, es una unidad un poco virtual (con ciertos comandos, un bloque puede ser de 1 KB o más).

Los caracteres b, c, w o k pueden aparecer tras el valor colocado después del criterio.

Carácter	Significado
B	Por defecto, si no se especifica, es un bloque de 512 bytes.
C	Es un carácter en el sentido ASCII; por lo tanto, 1 byte.
W	Es una palabra (en el sentido antiguo) de 2 bytes.
K	1 KB (1024 bytes).

El valor puede ir precedido de un + o un - que significan "más de" o "menos de". Sin esta indicación, el tamaño buscado debe corresponderse EXACTAMENTE.

- -size 5: busca los archivos de un tamaño de 5 bloques (512 bytes por bloque, o sea, 2560 bytes).
- -size 152c: busca los archivos de un tamaño de 152 caracteres (bytes).
- -size 10k: busca los archivos de un tamaño de 10 KB (10*1024 bytes = 10240 bytes).
- -size +5000k: los archivos de más de 5000 KB.

- -size -100k: los archivos de menos de 100 KB.

```
seb@slyserver:/var/log> find -size +100k
./zypper.log-20080227.bz2
./lastlog
./zypper.log-20080302.bz2
./wtmp
./zypper.log-20080226.bz2
./zypper.log
./messages
```

Se puede utilizar el criterio de búsqueda **-empty** en sustitución de -size 0.

e. -atime, -mtime y -ctime

- `-atime`: busca en la fecha del último acceso (access time). Un acceso puede ser la lectura del archivo, pero también el simple hecho de listarlo de manera específica.
- `-mtime`: busca en la fecha de la última modificación (modification time). Se trata de la modificación del contenido.
- `-ctime`: busca en la fecha de modificación (change time, en realidad la fecha de última modificación del número de inodo).

La fecha de modificación del archivo corresponde a la fecha en la cual se modificó por última vez la información relacionada con el inodo (ver capítulo Los discos y el sistema de archivos): modificación del nombre, desplazamiento, cambio de derechos, del tamaño, etc.

Estos tres criterios sólo trabajan con días (periodos de 24 horas). 0 es el mismo día; 1, ayer; 2, antes de ayer, etc. El valor n colocado después del criterio corresponde, por lo tanto, a n*24 horas. Este rango no es fijo, ya que "ayer" significa entre 24 y 48 horas...

Los signos + o - permiten precisar los términos "de más" y "de menos":

- -mtime 1: archivos modificados ayer (entre 24 y 48 horas).
- -mtime -3: archivos modificados hace menos de tres días (72 horas).
- -atime +4: archivos modificados hace más de cuatro días (más de 96 horas).

```
seb@slyserver:/var/log> find . -mtime -1
./kdm.log
./vmware
./vmware/vmware-serverd-0.log
./vmware/vmware-serverd.log
./mail.info
./Xorg.0.log
./lastlog
./Xorg.0.log.old
./warn
...
```

Eche un vistazo a los criterios **-newer**, **-anewer** y **-cnewer**, que toman un archivo como parámetro. En este caso, find busca los archivos que son más recientes que el designado.

f. -perm

`-perm` permite efectuar búsquedas en las autorizaciones de acceso (derechos, SUID, SGID, Sticky).

Se deben precisar los derechos en base 8 (valor octal) y completos. El carácter - colocado antes del valor octal significa que los archivos buscados deben tener al menos los permisos establecidos. En el ejemplo siguiente, se buscan los directorios sobre los cuales todo el mundo (user, group, others) tiene permiso de ejecución (permiso x, o sea 1).

```
seb@slyserver:/var/log> find -type d -perm -111
.
./vmware
./vmware/vmsd-xaction
./cups
```

g. -links y -inum

Aunque estos criterios se refieren a nociones más avanzadas del sistema de archivos, es buena idea presentarlos ahora. Podrá volver a ellos en cuanto el capítulo Los discos y el sistema de archivos le presente el funcionamiento interno de un sistema de archivos.

La opción `-links` permite una búsqueda por nombre de hard links. Puede precisar los signos + o - (más de n vínculos y menos de n vínculos). Un archivo normal único posee 1 vínculo. Un directorio, 2 vínculos (la entrada en el catálogo al cual pertenece y en el punto). Para una búsqueda de vínculos simbólicos, habrá que utilizar la opción `-type l`.

```
$ find . -type f -links +2 -print
./fic2
./hardlink3_fic2
./hardlink_fic2
./hardlink2_fic2
```

`-inum` permite una búsqueda por número de inodo. Es útil en el caso de una búsqueda de todos los vínculos que llevan un mismo número de inodo. El número de inodo puede verse mediante la opción `-i` del comando **ls**.

```
seb@slyserver:/var/log> ls -i
491891 acpid                491793 mail.info    491860 Xorg.0.log
491791 boot.log             491794 mail.warn    490686 Xorg.0.log.old
491729 boot.msg             492046 mcelog       492060 Xorg.1.log
seb@slyserver:/var/log> find . -inum 491791 -print
./boot.log
```

3. Comandos

Además de la opción `-print`, hay otras opciones que permiten efectuar una acción en los archivos encontrados.

a. -ls

El criterio muestra información detallada en los archivos encontrados que corresponden al criterio, en lugar del simple nombre de archivo. La salida corresponde a un comando **ls** con los parámetros`d`, `i`, `l` y `s` (tamaño en bloques de 1 KB).

```
seb@slyserver:~> find -size +500000k -ls
2342935 584388 -rw-r--r--   1 seb      users    597817344 feb 24
11:52 ./eeexubuntu-7.10.3-desktop-i386.iso
```

b. -exec

El criterio `-exec` va a ejecutar el comando colocado justo después con cada coincidencia encontrada. Es necesario hacer algunas observaciones:

- `-exec` debe ser obligatoriamente la última opción del comando **find**.
- El comando ejecutado por `-exec` debe terminarse con un ";". Este carácter especial debe escribirse con \; para que el shell no lo interprete.
- Con el fin de pasar como parámetro para el comando el archivo encontrado por find, hay que escribir {} (sustitución del archivo).

Ejemplo para borrar todos los archivos que terminan por ".mp3":

```
$ find . -type f -name "*.mp3" -exec rm -f {} \;
```

El sistema no espera la finalización del comando **find** para ejecutar el comando **rm**. Lo ejecuta en cuanto **find** encuentra un archivo. Así, si el comando anterior le ha mostrado n archivos antes de que pensara interrumpirlo, estos n archivos ya han sido borrados.

c. -ok

El criterio `-ok` es idéntico a la opción `-exec`, pero, para cada coincidencia, se le requiere una confirmación al usuario.

```
$ find . -inum 95 -ok rm -f {} \;
< rm ... ./fic1 > (yes)?   n
< rm ... ./vínculo_fic1 > (yes)?   y
```

4. Criterios AND / OR / NOT

Es posible combinar las opciones de criterio de selección. Si no se especifica ninguno, el Y lógico está implícito.

Criterio	Acción
-a, -and	AND, Y lógico, por defecto
-o, -or	OR, O lógico
!	Negación del criterio

Ejemplo con todos los archivos que no contienen fic en su nombre, y todos los archivos que no son ni normales ni directorios.

```
$ find . ! -name "*fic*" -print
.
./dir1
./lista
./mypass
./users
./lista2
./ls.txt
./pepito.tar.gz
./nohup.out
./lista_ls
./dir2
./seb1
./seb2
$ find . ! \( -type f -o -type d \) -ls
     409    0 lrwxrwxrwx  1 oracle   system          4 Ago 14 15:21
./vínculo_fic1 -> fic1
     634    0 lrwxrwxrwx  1 oracle   system          4 Ago 14 15:21
./vínculo_fic2 -> fic2
```

5. Encontrar ejecutables

a. whereis

El comando **whereis** busca en las rutas de archivos binarios, del manual y de las fuentes los archivos que corresponden a los criterios facilitados.

```
$ whereis date
date: /bin/date /usr/share/man/man1/date.1.gz
/usr/share/man/man1p/date.1p.gz
```

Puede precisar algunos parámetros:

- `-b` únicamente para los binarios,
- `-m` únicamente para los manuales,
- `-s` únicamente para las fuentes.

Se buscan los archivos por defecto en:

```
/{bin,sbin,etc}
/usr/{lib,bin,old,new,local,games,include,etc,src,man,sbin,X386,TeX,
g++-include}
/usr/local/{X386,TeX,X11,include,lib,man,etc,bin,games,emacs}
```

Por lo tanto, no se sorprenda de obtener lo siguiente:

```
$ whereis -b passwd
passwd: /usr/bin/passwd /etc/passwd /etc/passwd.old
/etc/passwd.YaST2save /etc/passwd.vipwKSnTgH /usr/bin/X11/passwd
```

b. which

El comando **which** busca un comando en el PATH (ruta de los ejecutables) y le facilita el primero que encuentra:

```
$ which date
/bin/date
```

En Linux hay comandos con el mismo nombre en varias rutas. Debe precisar el parámetro `-a` para que `which` siga su búsqueda. Sin embargo, si usted ejecuta el comando sin especificar una de esas rutas, se ejecutará el primero por defecto.

```
$ which -a passwd
/usr/bin/passwd
/usr/bin/X11/passwd
```

c. locate

El comando **locate** busca un archivo según el modelo dado en una base de datos de archivos construida por el comando **updatedb**.

El comando **updatedb** recorre una serie de rutas en las que ejecuta un **find** y almacena todos los resultados en una base indexada. Esto evita, por lo tanto, efectuar de nuevo un **find** para las búsquedas clásicas. En la práctica, basta con pasarle a **updatedb** la lista de las rutas o no incluir los archivos en la base indexada.

contrab suele lanzar **updatedb** a diario. A veces se colocan los parámetros del comando en un

archivo **/etc/sysconfig/locate**.

```
$ cat /etc/sysconfig/locate | grep -Ev "^(#|$)"
RUN_UPDATEDB=yes
RUN_UPDATEDB_AS=nobody
UPDATEDB_NETPATHS=""
UPDATEDB_PRUNEPATHS="/mnt /cdrom /tmp /usr/tmp /var/tmp /var/spool
/proc /media /sys"
UPDATEDB_NETUSER=""
UPDATEDB_PRUNEFS=""
```

En este caso, se ejecuta el comando siguiente:

```
# updatedb --localuser=nobody --prunepaths=/mnt /cdrom /tmp /usr/tmp
/var/tmp /var/spool /proc /media /sys
```

Si usted ejecuta el comando de esta manera, puede que éste consuma todos los recursos de procesador de su máquina. Por eso, se ejecuta **updatedb** con una prioridad baja.

Se coloca la base de datos en **/var/lib/locatedb**.

```
$ locate pepito
/opt/kde3/share/apps/ksgmltools2/docbook/xsl/html/auptotoc.xsl
/opt/kde3/share/apps/ksgmltools2/docbook/xsl/params/autotoc.label
.separator.xml
/usr/share/gnome/help/gnome-doc-xslt/C/db2html-autotoc.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/fo/autotoc.xsl
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/html/autotoc.xsl
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/params/autotoc.label
.in.hyperlink.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/params/autotoc.label
.separator.xml
/usr/share/xml/docbook/stylesheet/nwalsh/1.73.1/xhtml/autotoc.xsl
/usr/share/xml/gnome/xslt/docbook/html/db2html-autotoc.xsl
/usr/src/linux-2.6.22.17-0.1/drivers/mtd/maps/omap-pepito-flash.c
/usr/src/linux-2.6.22.17-0.1/drivers/mtd/nand/pepito.c
/usr/src/linux-2.6.24.4/drivers/mtd/maps/omap-pepito-flash.c
/usr/src/linux-2.6.24.4/drivers/mtd/nand/pepito.c
```

El editor vi

1. Presentación

El editor de Unix por defecto se llama **vi** (*visual editor*). No es de los más ergonómicos en comparación con editores en modo gráfico, pero tiene la ventaja de estar disponible y de utilizar la misma sintaxis básica en todos los Unix. Cada Unix propone en general una sintaxis extendida más allá de la sintaxis básica. El editor vi en Linux se llama **vim**. Vim respeta toda la sintaxis de vi, pero no es recíproco. Vi es pequeño: ocupa poco espacio en disco, consume poca memoria. Algunas de las opciones disponibles sólo funcionan en vim, como por ejemplo la gestión de buffers de memoria múltiples.

```
vi [opciones] Archivo [Archivo2 ...]
```

vi no tiene menús, tampoco interfaz gráfica, y no es intuitivo. Requiere conocerse de memoria un determinado número de atajos de teclado para poder utilizarlo. Si bien el aprendizaje es un poco difícil, una vez dominado, vi es rápido y práctico, hasta tal punto de ir más rápido que con editores de texto gráficos.

El debate que enfrenta a los partidarios de emacs (o de otros editores) con los de vi no tiene razón de ser. Cualquier sistema Linux (y Unix) dispone pase lo que pase del editor vi, por lo que es ineludible. Si tiene la oportunidad, puede instalar el editor que le guste, aunque esto, en entornos empresariales, de servidores, etc., no siempre será posible.

2. Funcionamiento

Existen tres modos de funcionamiento:

- modo **comando**: las inserciones representan comandos. Se accede a ellos al pulsar [Esc]. Cada tecla o combinación de teclas activa una acción (supresión de líneas, inserciones, mover, copiar, pegar, etc.).
- modo **inserción**: se trata de la inserción de texto clásico.
- modo **línea de comandos**: una línea en la parte inferior de la pantalla permite insertar comandos especiales, validada con la tecla [Entrar]. Se accede a ella al presionar, en modo comando, la tecla ":".

Cuando ejecuta vi, por defecto está en modo comando. Para empezar a introducir texto hay que teclear un comando de añadido o inserción, lo que enerva rápidamente cuando uno no conoce vi: a o i. Por lo tanto, el comando [Esc] **i** permite insertar texto. Para salir, puede pasar por el modo línea de comandos. Teclee [Esc]: y luego q y [Entrar]. ¡Es sencillo!

Si ha usado un comando ya se encuentra en modo comando y no tiene que pulsar la tecla [Esc] de nuevo para enviar el siguiente. Por ejemplo, la "x" suprime un carácter. Si ya está en modo comando, puede pulsar 10 veces la "x" sin darle a [Esc] entre cada pulsación.

Por supuesto, hay un punto de ironía. Nadie conoce todos los atajos; el propio autor de este libro no los recuerda todos. Por lo tanto, además de las guías en la prensa especializada, aquí tiene la lista de los comandos más utilizados en forma de tablas. Cuando el ":" precede al comando, éste se debe insertar en línea de comandos.

Si se pierde, en cualquier caso y pase lo que pase, al pulsar la tecla [Esc] vuelve siempre en modo comando.

3. Los comandos

a. La introducción de datos

Se deben efectuar las acciones siguientes en modo comando. Antes se debe pulsar [Esc]: [Esc] a, [Esc] i, etc.

Comando	Acción
a	Añadir después del carácter actual.
A	Añadir texto al final de línea.
i	Insertar delante del carácter actual, como en un procesador de texto.
I	Insertar texto al principio de línea.
o	Añadir una línea debajo de la línea actual.
O	Insertar una línea encima de la línea actual.

b. Salir y guardar

Recuerde que ":" significa que el comando se introduce en la línea de comandos: [Esc] :, inserción del comando, luego [Entrar].

Comando	Acción
ZZ	Guardar el archivo y sale.
:q!	Salir sin guardar.
:q	Salir si no se modificó el archivo (en caso contrario, aparece un mensaje de error).
:w	Guardar el archivo. Puede especificar un nombre para seguir.
:wq o :x	Guardar y salir.
1,10w fic	Guardar las líneas de 1 a 10 en fic.

c. Desplazamiento

Quizá la idea le haga sonreír, pero existen todavía teclados sin teclas direccionales, así como programas e intérpretes de comando que no saben interpretarlos (ejemplo del shell ksh mal configurado en algunos Unix). En estas máquinas, se activa el modo de inserción de vi por defecto e, incluso en vi, es posible arreglárselas sin las teclas de dirección, en modo comando.

Comando	Acción
h	Ir a la izquierda.
l (l minúscula)	Ir a la derecha.
k	Ir hacia arriba.
j	Ir hacia abajo.
0 (cero)	Principio de línea.
:0	Principio de archivo (primera línea).
$	Fin de línea.
:$	Fin de archivo (última línea).
w	Ir a la palabra siguiente.

b	Ir a la palabra anterior.
f<c>	Saltar al carácter <c> siguiente.
Control+f	Avanzar una pantalla (Forward).
Control+b	Retroceder una pantalla (Backward).
G	Última línea del archivo.
<n>G	Salta a la línea "n" (p. ej.: 10G va a la décima línea).
:<n>	Ídem (:10 va a la décima línea).

d. La corrección

Comando	Acción
x	Borrar el carácter bajo el cursor.
X	Borrar el carácter delante del cursor.
r<c>	Sustituir el carácter bajo el cursor por el carácter <c>.
dw	Borrar desde el cursor hasta el final de la palabra.
d$ o **D**	Borrar desde el cursor hasta el final de la línea.
dO	Borrar desde el principio de línea hasta el cursor.
df<c>	Borrar todo hasta el carácter <c>.
dG	Borrar todo hasta la última línea, incluida la línea actual.
d1G	Borrar todo hasta la primera línea, incluida la línea actual.
dd	Borrar la línea actual.
u	Undo. Cancelar la última acción.

Se pueden repetir estos comandos. 5dd suprime 5 líneas; 4dw, 4 palabras; 5x, 5 caracteres, etc.

e. Búsqueda en el texto

A diferencia de un editor de texto clásico, vi puede buscar otras cosas además de palabras sencillas, y funciona con la ayuda de caracteres especiales y criterios. El comando de búsqueda es el carácter /. La búsqueda arranca en el carácter actual hasta el final del archivo. El carácter ? efectúa la búsqueda en sentido inverso. Primero se indica el criterio y luego se pulsa [Entrar].

```
/echo
```

busca la cadena 'echo' a continuación en el archivo. Cuando se encuentra la cadena, el cursor se para en el primer carácter de esta cadena.

El comando n permite seguir la búsqueda en el sentido indicado al principio. El comando N efectúa la búsqueda en sentido inverso.

Algunos criterios

- /[CcBb]ola: Cola, cola, Bola, bola.
- /[A-Z]e: todo lo que empieza con una mayúscula con una e en segunda posición.
- /[A-Za-Z0-9]: todo lo que empieza por una mayúscula, una minúscula o una cifra.
- /[^a-z]: intervalo negativo: todo lo que no empieza con una minúscula.
- /co.a: el punto sustituye un carácter: coma, cola, cota...

- /Au*o: la estrella es un carácter de repetición, de 0 a n caracteres, Auo, Auto, Automoto...
- /.* : la estrella delante del punto, una cadena cualquiera de tamaño variable; el "." representa un carácter.
- /[A-Z][A-Z]*: repetición del motivo entre [] de 0 a n veces, búsqueda de una palabra que tiene al menos una mayúscula (al principio de palabra).
- /^Auto: el ^ indica que la cadena buscada deberá estar al principio de línea.
- /Auto$: el $ indica que la cadena buscada deberá estar al final de línea.

f. Comandos de sustitución

Para sustituir texto, hay que colocarse al principio de la cadena que desea modificar y teclear uno de los comandos siguientes. A continuación, teclee simplemente su texto. El principio es sencillo. Es idéntico al de la supresión: se suprime la parte indicada y vi pasa al modo de inserción para la introducción de datos.

Comando	Acción
cw	Sustituir la palabra corriente.
c$	Sustituir hasta el final de la línea.
c0 (cero)	Sustituir hasta el principio de la línea.
cf<x>	Sustituir hasta el próximo carácter <x>.
c/<rech>	Sustituir hasta la próxima coincidencia de la cadena <rech>.

g. Copiar - Pegar

El comando **v** permite una selección visual. Se subraya el texto y se puede desplazar el cursor para seleccionar el texto. Luego utilice uno de los comandos siguientes:

- el comando "d" corresponde a cortar (mover);
- la c hace casi lo mismo, pero vi se queda en modo edición;
- para pegar el texto en el lugar elegido, se utiliza el comando p (detrás del carácter) o P (delante del carácter). Si se ha copiado una línea entera, se colocará debajo de la línea activa.

Las acciones siguientes son posibles en modo comando:

- Para copiar una línea: yy.
- Para copiar cinco líneas: 5yy.
- Para colocar las líneas copiadas en un sitio determinado: p.
- El editor vi dispone de 26 búferes de memoria para almacenar en ellos los datos que se pueden nombrar como se desee. Para ello, se utiliza el ".
- Para copiar cinco palabras en la memoria m1: "m1y5w.
- Para pegar el contenido de la memoria m1 en un lugar determinado: "m1p.

h. Sustitución

La sustitución permite reemplazar de manera automática varias coincidencias por otra cadena.

```
:[1ª línea, última línea]s/Modelo/Sustitución/[gc]
```

Los números de líneas son opcionales. En este caso, la sustitución se hace en la línea corriente

únicamente. Como sustitución de los números de línea, . determina la línea corriente, **1** la primera línea, **$** la última línea.

El modelo es un criterio de búsqueda presentado en estas últimas páginas. Sustitución es una cadena cualquiera que sustituirá el modelo.

Por defecto sólo se sustituye la primera coincidencia. La letra g indica que hay que sustituir todas las coincidencias. Con c, vi requiere una confirmación para cada una de las coincidencias. El siguiente ejemplo sustituye todas las coincidencias de Unix o unix en UNIX.

```
:1,$s/[Uu]nix/UNIX/g
```

i. Otros

Edición avanzada

Presentamos algunos comandos prácticos.

:r fic	Insertar el contenido de fic desde el lugar actual.
:! cmd	Ejecutar el comando. Pulse [Entrar] para volver a vi.
:r! cmd	Insertar el resultado del comando en el lugar actual.
:e fic	Cargar el archivo fic para edición.
:e#	Conmutar entre los diferentes archivos abiertos.

Comando set

El comando **set** permite configurar el editor y acceder a sus opciones:

- set all: muestra el conjunto de las opciones posibles.
- set number (o nu) / nonumber (o nonu): muestra/suprime los números de líneas.
- set autoindent / noautoindent: se conserva el sangrado durante un retorno a la línea.
- set showmatch / noshowmatch: durante la introducción de una llave o de una paréntesis de cierre, se visualiza el de apertura durante un instante muy corto, luego el editor vuelve al carácter corriente.
- set showmode / noshowmode: vi mostrará una línea de estado (INPUT MODE).
- set tabstop=x: define el número de caracteres para una tabulación.

Vim propone a menudo la coloración sintáctica. Detecta el tipo de archivo cargado y colorea las líneas, las contraseñas, etc. Puede desactivar la coloración:

- syntax off para desactivar;
- syntax on para activar.

Redirecciones

1. Fundamentos

Las redirecciones son una de las más importantes posibilidades proporcionadas por el shell. Por redirección se entiende la posibilidad de redireccionar la visualización de la pantalla hacia un archivo, una impresora o cualquier otro periférico, los mensajes de errores hacia otro archivo, de sustituir la introducción vía teclado por el contenido de un archivo.

Cualquier flujo de datos en entrada o salida de comando pasa por un canal. Como sucede con el agua, es posible desviar el curso de los datos hacia otro destino o desde otra fuente.

Linux utiliza canales de entradas/salidas para leer y escribir sus datos. Por defecto, el canal de entrada es el teclado, y el canal de salida, la pantalla. Los errores, direccionados por defecto a la pantalla, son tratados como un canal especial.

Es posible redireccionar estos canales hacia archivos o flujo de texto de manera transparente para los comandos Linux.

2. De salida

Se puede utilizar el carácter **>** para redireccionar la salida estándar (la que va normalmente en la pantalla). Luego se indica el nombre del archivo donde se colocarán los resultados de salida.

```
$ ls -l > resultado.txt
$ cat resultado.txt
total 1
-rw-r--r--    1 Administ ssh_user        0 Jul  4 12:04 PEPITO
-rw-r--r--    1 Administ ssh_user        0 Jul 25 15:13 resultado.txt
-rw-r--r--    1 Administ ssh_user      171 Jul 25 15:13 test.txt
```

Si no existe, se creará el archivo. Si existe, se sobreescribirá su contenido, incluso si el comando tecleado no es correcto. **El shell empieza primero por crear el archivo y luego ejecuta el comando**.

Es un aspecto importante de las redirecciones: se interpretan las redirecciones de derecha a izquierda, y se instalan las redirecciones ANTES de la ejecución de los comandos. Hay que crear el archivo antes de poder escribir en él. De ahí que, incluso si el comando es falso, se crea o se "chafa" el archivo...

Para añadir datos a continuación del archivo, o sea, sin sobreescribirlos, se utiliza la doble redirección >>. Se añade el resultado del comando al final del archivo.

```
$ ls -l > resultado.txt
$ date >> resultado.txt
$ cat resultado.txt
total 1
-rw-r--r--    1 Administ ssh_user        0 Jul  4 12:04 PEPITO
-rw-r--r--    1 Administ ssh_user        0 Jul 25 15:13 resultado.txt
-rw-r--r--    1 Administ ssh_user      171 Jul 25 15:13 test.txt
Thu Jul 25 15:20:12  2002
```

3. En entrada

Los comandos que esperan datos o parámetros desde el teclado pueden también recibirlos desde un archivo usando el carácter <. Un ejemplo con el comando **wc** (*word count*), que permite contar el número de líneas, de palabras y de caracteres de un archivo:

```
$ wc < resultado.txt
     4      29     203
```

4. Documento en línea

La redirección << es algo particular. Permite la utilización de los documentos en línea. Encontrará a veces el término Herescript o Here Document. Esto permite la inserción de un texto hasta un punto dado y el envío de su resultado a un comando o un filtro. Se autorizan las redirecciones clásicas. Después del <<, ha de indicar una cadena que define el final de la inserción; por ejemplo, aquí 'end'.

```
$ tr "[a-z]" "[A-Z]" << end
> hola amigos
> esto es un ejemplo
> de herescript
> end
HOLA AMIGOS
ESTO ES UN EJEMPLO
DE HERESCRIPT
```

5. Los canales estándares

Se puede considerar un canal como un archivo que posee su propio descriptor por defecto, y en el cual se puede leer o escribir.

- El canal de entrada estándar se llama **stdin** y lleva el descriptor 0.
- El canal de salida estándar se llama **stdout** y lleva el descriptor 1.
- El canal de error estándar se llama **stderr** y lleva el descriptor 2. Se puede redireccionar el canal de error hacia otro archivo.

```
$ rmdir directorio2
rmdir: `directorio2': No such file or directory
$ rmdir directorio2 2>error.log
$ cat error.log
rmdir: `directorio2': No such file or directory
```

Puede redireccionar los dos canales de salida a un único archivo poniéndolos en relación. Para ello, se utiliza el **>&**. También es importante saber en qué sentido el shell interpreta las redirecciones. El shell busca primero los caracteres **<**, **>**, **>>** al final de la línea, ya que las redirecciones suelen estar al final de comando. Por lo tanto, si quiere agrupar los dos canales de salida y de error en un mismo archivo, hay que proceder como a continuación.

```
$ ls -l > resultado.txt 2>&1
```

Se redirecciona la salida 2 hacia la salida 1; por lo tanto, los mensajes de error pasarán por la salida estándar. Luego se redirecciona el resultado de la salida estándar del comando **ls** hacia el archivo resultado.txt. Este archivo contendrá, por lo tanto, a la vez la salida estándar y la salida de error.

Puede utilizar los dos tipos de redirección a la vez:

```
$ wc < resultado.txt > cuenta.txt
$ cat cuenta.txt
      4      29     203
```

6. Apertura de canales

Existen tres canales estándares y se numeran de 0 a 2. Así, 0< equivale a < y 1> a >. El comando**exec** permite abrir otros siete canales numerados de 3 a 9. Por lo tanto, hay diez canales en total.

Puede e incluso debe considerar, en el marco de los procesos, sacar algunos resultados por el canal 3, otros por el 4, y así sucesivamente. Los canales están abiertos para la entrada y la salida.

```
$ exec 3>dump.log
$ ls -l >&3
$ cat dump.log
total 3952
-rw-r--r--  1 seb users  167212 oct  9 09:27 battlestar_1280.jpg
drwxr-xr-x  2 seb users    4096 mar  4 08:51 bin
drwxr-xr-x  8 seb users    4096 mar  4 08:45 cxoffice
drwx------  2 seb users    4096 mar 10 12:29 Desktop
drwx------ 13 seb users    4096 mar  6 11:49 Documents
-rw-r--r--  1 seb users       0 mar 11 11:34 dump.log
-rw-r--r--  1 seb users 3785296 dic 12 15:15 e3555_EeePC4G.pdf
drwxr-xr-x  3 seb users    4096 mar 10 11:16 Games
drwxr-xr-x  5 seb users    4096 mar 10 11:16 karchiver-3.4.2.b4
-rw-r--r--  1 seb users     358 mar 11 08:51 lista
-rw-r--r--  1 seb users     608 mar 11 09:14 tmpgrp
-rw-r--r--  1 seb users    1555 mar 11 09:15 tmppwd
```

Todo lo que se escribirá en el canal 3 se escribirá en el archivo dump.log. Luego se puede cerrar el canal relacionándolo con un seudocanal (canal de cierre -).

```
$ exec 3>&-
```

7. Filtro: definición

Un **filtro** (o un comando filtro) es un programa que sabe escribir y leer datos por los canales estándares de entrada y salida. Modifica o trata si es preciso el contenido. **wc** es un filtro. Le presentamos algunos: **more** (muestra los datos página por página), **sort** (ordena los datos), **grep**(criterios de búsqueda).

8. Pipelines/tuberías

Las redirecciones de entrada/salida como las que acaba de ver permiten redireccionar los resultados hacia un archivo. Luego se puede reinyectar en un filtro para extraerle otros resultados. Esto obliga a teclear dos líneas: una para la redirección hacia un archivo, otra para redireccionar este archivo hacia el filtro. Las **tuberías** o **pipes** permiten redirigir el canal de salida de un comando hacia el canal de entrada de otro. El carácter que lo permite | está accesible con la combinación [AltGr] **1** de los teclados españoles.

```
$ ls -l > resultado.txt
$ wc < resultado.txt
```

se convierte en

```
$ ls -l | wc
```

Es posible colocar varios | en una misma línea.

```
$ ls -l | wc | wc
      1       3      24
```

El primer comando no tiene por qué ser un filtro. No es el caso más habitual. Lo importante es que se debe facilitar un resultado. Ídem para el último comando, que puede ser por ejemplo un comando de edición o impresión. Finalmente, el último comando puede ser objeto de una redirección de salida.

```
$ ls -l | wc > resultado.txt
```

Filtros y herramientas

Un **filtro** (o un comando filtro) es un programa que sabe escribir y leer datos por los canales estándares de entrada y salida. Modifica o trata si es preciso el contenido. **wc** es un filtro. Las herramientas no siempre se comportan como filtros. Permiten un determinado número de acciones en archivos y su contenido, como, por ejemplo, dar formato o imprimir.

1. Extracción de los nombres y rutas

El comando **basename** permite extraer el nombre del archivo en una ruta.

```
$ basename /tmp/seb/lista
lista
```

El comando **dirname** efectúa lo contrario, extrae la ruta.

```
$ dirname /tmp/seb/lista
/tmp/seb
```

2. Búsqueda de líneas

Se trata de extraer líneas de un archivo según varios criterios. Para ello, dispone de tres comandos:**grep**, **egrep** y **fgrep**, que leen los datos o bien desde un archivo de entrada, o bien desde el canal de entrada estándar.

a. grep

La sintaxis del comando **grep** es

```
grep [Opciones] modelo [Archivo1...].
```

El modelo se compone de criterios de búsqueda que se parecen mucho a los criterios ya expuestos para vi, por ejemplo. No hay que olvidar que se debe interpretar estos criterios con el comando**grep**, y no con el shell. Por lo tanto, hace falta cerrar todos los caracteres.

```
$ cat fic4
Cerdo
Ternera
Buey
rata
Rata
buey
$ grep "^[bB]" fic4
Buey
buey
```

El comando **grep** también puede tomar algunas opciones interesantes.

- **-v** efectúa la búsqueda inversa: se visualizan todas las líneas que no corresponden a los criterios.
- **-c** sólo devuelve el número de líneas encontradas, sin mostrarlas.
- **-i** no diferencia las mayúsculas de las minúsculas.
- **-n** indica el número de línea para cada línea encontrada.
- **-l** en el caso de archivos múltiples, indica en qué archivo se ha encontrado la línea.

```
$ grep -i "^b" fic4
Buey
buey
```

b. egrep

El comando **egrep** extiende los criterios de búsqueda y puede aceptar un archivo de criterios en entrada. Equivale a un grep -E. Emplea como criterios expresiones regulares.

```
egrep -f archivo_criterio archivo_búsqueda
```

Carácter especial	Significado
|	O lógico, la expresión colocada antes o después debe desaparecer.
(...)	Agrupación de caracteres.
[...]	Un carácter tiene esta posición entre los indicados.
. (punto)	Cualquier carácter.
+	Repetición, el carácter colocado antes debe aparecer al menos una vez.
*	Repetición, el carácter colocado antes debe aparecer de cero a n veces.
?	El carácter colocado antes debe aparecer una vez como máximo.
{n}	El carácter colocado antes debe aparecer exactamente n veces.
{n,}	Aparece n veces o más.
{n,m}	Aparece entre n y m veces.
^	En principio de cadena.
$	En final de cadena.

Únicamente "buenas tardes" y "buenas noches" empezarán por una mayúscula o una minúscula si están solos en una línea:

```
^[bB]uenas(tardes|noches)$
```

Verificación muy escueta de la validez de una dirección IP:

```
echo $IP | egrep '([0-9]{1,3}\.){3}[0-9]{1,3}'
```

Esta línea se descompone de la manera siguiente:

- ([0-9]{1,3}\.){3}: www.xxx.yyy.
  - [0-9]: un carácter entre 0 y 9
  - {1,3}: repetido entre una y tres veces, por lo tanto: x, xx o xxx
  - \.: seguido de un punto
  - {3}: el conjunto tres veces

Luego [0-9]{1,3}: .zzz

- [0-9]: un carácter entre 0 y 9
- {1,3}: repetido entre una y tres veces

c. fgrep

El comando **fgrep** es un grep simplificado y rápido (fast grep) y equivale a un grep -F. Acepta también un archivo de criterios de búsqueda, pero debe tratarse de criterios simples, sin caracteres especiales. Introduzca en el archivo de criterios líneas sencillas (texto y cifras), una búsqueda por línea. Fgrep va a buscar en un archivo meta o un flujo en entrada las líneas que corresponden a cada uno de los criterios.

d. sed

El aprendizaje de sed requeriría todo un libro. Sed es un editor de flujo (Stream Editor) que permite filtrar y transformar texto. Es un poco como un editor que permite modificar texto vía comandos de scripts, pero en un paso y sin edición interactiva. Utiliza un juego extendido de comandos procedente del editor ed. Su sintaxis básica es:

```
sed -e '<cmd>' arch
```

Para utilizar sed, hay que aprender y entender las expresiones regulares. El cuadro del comando**egrep** retoma la sintaxis básica de las expresiones. Cualquier libro sobre sed parte de estas expresiones y recíprocamente.

Sed se utiliza muy a menudo para sustituir valores por otros (sustitución) o suprimir líneas particulares (aunque se podría utilizar grep en este caso). La sintaxis básica de sustitución es la siguiente:

```
s/<antiguo>/<nuevo>/[g]
```

La g final permite realizar una sustitución en toda la línea en caso de haya varias coincidencias. Aquí tiene un ejemplo que sustituye __NOMBRE__ por Pepito:

```
$ echo "Me llamo __NOMBRE__. ¿Te llamas __NOMBRE__?" | sed -e 's/__
NOMBRE__/Pepito/'
Me llamo Pepito. ¿Te llamas __NOMBRE?
$ echo "Me llamo __NOMBRE__. ¿Te llamas __NOMBRE__?" | sed -e 's/__
NOMBRE__/Pepito/g'
Me llamo Pepito. ¿Te llamas Pepito?
```

Puede colocar un valor numérico en el campo nuevo para precisar, si la búsqueda consta de varios elementos agrupados por paréntesis, en qué elemento de los buscados debe trabajar. Aquí tenemos un ejemplo sencillo que añade asteriscos alrededor del nombre pepito:

```
$ echo pepito | sed -e "s/\(pepito\)/**\1**/"
**pepito**
```

Para suprimir todas las líneas vacías o que contienen únicamente espacios:

```
$ sed -e '/^ *$/d'  archivo
```

3. Columnas y campos

El comando **cut** permite seleccionar columnas y campos en un archivo.

a. Columnas

La sintaxis es la siguiente:

```
cut -cColumnas [fic1...]
```

Una columna es la posición de un carácter en la línea. El primer carácter es la columna 1; el segundo, la columna 2, y así sucesivamente. Una línea de 80 caracteres dispone de 80 columnas. La numeración empieza en 1. Es el método ideal para archivos planos y con formato fijo, donde cada campo empieza y termina con posiciones dadas.

El formato de selección de columna es el siguiente:

- una columna sola (p. ej. -c2 para la columna 2);
- un intervalo (p. ej. -c2-4 para las columnas 2, 3 y 4);
- una lista de columnas (p. ej. -c1,3,6 para las columnas 1, 3 y 6);
- los tres a la vez (p. ej. -c1-3,5,6,12-).

```
$ cat lista
Producto  precio  cantidades
ratón     30      15
disco     100     30
pantalla  300     20
teclado   45      30

$ cut -c1-5 lista
Produ
ratón
disco
panta
tecla

$ cut -c1-3,10-12,15
Prorx   cantidades
rat0    15
dis0    30
pan0    20
tec530
```

b. Campos

El comando **cut** también permite seleccionar campos. Se deben delimitar estos campos por defecto por una tabulación, pero el parámetro `-d` permite seleccionar otro carácter (espacio, ;). La selección de los campos es idéntica a la de las columnas.

El carácter separador debe ser único. No es posible poner ni dos, ni tres, ni una cadena de separadores. Para eliminar los caracteres múltiples, utilice tr. Asimismo, el separador por defecto es la tabulación. Ahora bien, por defecto se sustituyen las tabulaciones con espacios dentro de los editores...

```
cut -dc -fCampos [fic1...]
```

Le presentamos unos ejemplos. El archivo lista contiene campos separados por tabulaciones.

```
$ cat lista
Producto  precio  cantidades
ratón     30      15
duro      100     30
disco     100     30
pantalla  300     20
teclado   45      30
tarjeta   45      30

$ cut -f1 lista
Producto
```

```
ratón
duro
disco
pantalla
teclado
tarjeta

$ cut -f1,3 lista
Producto cantidades
ratón    15
duro     30
disco    30
pantalla 20
teclado  30
tarjeta  30
```

Observe que, si invierte el orden de los campos (-f3,1), no obtendrá el efecto deseado: los campos salen siempre en el sentido 1,3.

A continuación vemos cómo aislar los nombres de un grupo y sus identificadores respectivos:

```
$ cat /etc/group
seb@slyserver:~> cat /etc/group
at:!:25:
audio:x:17:
avahi:!:106:
beagleindex:!:107:
bin:x:1:daemon
cdrom:x:20:
console:x:21:
daemon:x:2:
dialout:x:16:seb,esteban,enrique,public
disk:x:6:

$ cut -d: -f1,3 /etc/group
at:25
audio:17
avahi:106
beagleindex:107
bin:1
cdrom:20
console:21
daemon:2
dialout:16
disk:6
```

Si no hay delimitador (tabulación u otro) en una línea, **cut** muestra toda la línea.

4. Recuento de líneas

El comando **wc** (*word count*) permite contar las líneas, las palabras y los caracteres.

```
wc [-l] [-c] [-w] [-w] fic1
```

- `-l`: cuenta el número de líneas
- `-c`: cuenta el número de bytes
- `-w`: cuenta el número de palabras

- `-m`: cuenta el número de caracteres

```
$ wc lista
        12        48       234 lista
```

El archivo lista contiene 12 líneas, 48 palabras y 234 caracteres.

5. Ordenación de líneas

El comando **sort** permite ordenar las líneas. Por defecto, la ordenación se hace sobre toda la tabla en orden creciente. La ordenación es posible a partir de uno o varios campos. El separador de campos por defecto es la tabulación o, al menos, un espacio. Si hay varios campos, el primero es el separador; los demás son caracteres del campo.

La sintaxis de sort ha evolucionado desde hace varios años y Linux ha aplicado un estándar. Además, ya no utiliza la antigua sintaxis basada en +/-. En su lugar, hay que utilizar el parámetro `-k`. La numeración de los campos empieza con 1.

```
sort [opciones] [-k pos1[,pos2]] [fic1...]

$ cat lista
ratón      óptico  30      15
duro       30giga  100     30
duro       70giga  150     30
disco      zip     12      30
disco      blando  10      30
pantalla   15      150     20
pantalla   17      300     20
pantalla   19      500     20
teclado    105     45      30
teclado    115     55      30
tarjeta    sonido  45      30
tarjeta    vídeo   145     30
```

A continuación vemos cómo ordenar por orden alfabético a partir de la primera columna:

```
$ sort -k 1 lista
disco      blando  10      30
disco      zip     12      30
duro       30giga  100     30
duro       70giga  150     30
pantalla   15      150     20
pantalla   17      300     20
pantalla   19      500     20
ratón      óptico  30      15
tarjeta    sonido  45      30
tarjeta    vídeo   145     30
teclado    105     45      30
teclado    115     55      30
```

Algunos parámetros

Opción	Función
-d	Dictionnary sort (ordenación de diccionario). Sólo toma como criterio de ordenación las letras, las cifras y los espacios.
-n	Ordenación numérica, ideal para las columnas de cifras.
-b	Ignora los espacios al principio del campo.

-f	No hay diferencias entre mayúsculas y minúsculas (conversión en minúsculas y luego ordenación).
-r	Reverse, ordenación en orden decreciente.
-tc	Nuevo delimitador de campo c.

Ejemplo: ordenación numérica a partir de los precios por productos en orden decreciente:

```
$ sort -n -r -k 3 lista
pantalla   19      500     20
pantalla   17      300     20
pantalla   15      150     20
duro       70giga  150     30
tarjeta    vídeo   145     30
duro       30giga  100     30
teclado    115     55      30
teclado    105     45      30
tarjeta    sonido  45      30
ratón      óptico  30      15
disco      zip     12      30
disco      blando  10      30
```

También es posible ejecutar la ordenación desde un determinado carácter de un campo. Para ello, debe especificar el ".pos": -k1.3 empezará la ordenación a partir del tercer carácter del campo 1.

6. Eliminación de las líneas repetidas

El comando **uniq** permite suprimir las líneas repetidas en flujos de entrada o archivos ordenados. Por ejemplo, a continuación se muestra cómo sacar únicamente la lista de los GID realmente utilizados como grupo principal de usuarios:

```
$ cut -d: -f4 /etc/passwd | sort -n | uniq
0
1
2
7
8
12
13
14
25
49
51
62
...
```

7. Unión de dos archivos

a. En los campos comunes

El comando **join** permite efectuar la unión de dos archivos en función de un campo común. Se deben ordenar los dos archivos en los campos especificados en la unión.

```
join [-tc] [-1 n] [-2 m] fic1 fic2
```

La opción **-t** indica los separadores, **-1** el campo del primer archivo y **-2** el campo del segundo archivo, en los cuales efectuar la unión. Observe que **join** gestiona mal los duplicados y puede que se detenga como consecuencia de ello.

Es posible que el comando **join** no le proporcione el resultado esperado. La razón es que se detiene en cuanto no encuentra correspondencia entre dos líneas.

b. Línea a línea

El comando **paste** agrupa n archivos en uno. Para ello, concatena las líneas de cada uno de los archivos en una sola línea: línea1 de fic1 con línea2 de fic2, línea3 de fic 3, y así sucesivamente. Es un poco lo contrario de cut. El separador por defecto es la tabulación, pero puede precisar un delimitador con `-d`.

```
$ cat fic1
lista_a
lista_b
lista_c

$ cat fic2
lista_a2
lista_b2
lista_c2

$ paste -d: fic1 fic2
lista_a:lista_a2
lista_b:lista_b2
lista_c:lista_c2
```

8. División de un archivo en partes

a. Recortar

Aquí tenemos un comando muy práctico, **split**, que permite recortar un gran archivo en varios trozos con un tamaño determinado. Los sistemas de archivos no son todos iguales frente al tamaño máximo de un archivo. En Linux, el problema no es habitual, ya que un sistema de archivos de tipo ext3 puede soportar archivos de 1 TB (TB = TeraByte; 1 Terabyte = 1024 GB), o sea, el equivalente a unos 130 DVD de doble capa. Pero las bandas magnéticas, o en menor medida los discos removibles, no disponen de esta posibilidad.

Se suelen formatear los pendrives o un disco externo con un sistema de archivos de tipo VFAT, procedente del mundo de Microsoft. Este sistema de archivos, que procede de DOS y luego de Windows 9x, garantiza una compatibilidad entre todos los sistemas (Unix, Windows, Mac OS), ya que quien puede lo más, puede lo menos. VFAT (o más bien FAT16 o FAT32) soporta únicamente archivos de un tamaño máximo de 4 GB. Una imagen ISO de DVD o una carpeta de copia de seguridad no puede entrar en un solo bloque. Por lo tanto, hace falta dividir el archivo en varias partes.

```
split [-l n] [-b n[bkm] [archivo [prefijo]]
```

El comando puede funcionar según dos modos:

- recorte por líneas con `-l`: los archivos de salida tendrán todos n líneas de texto (salvo el último si se da el caso);
- recorte a tamaño fijo con `-b`: los archivos tendrán todos un tamaño fijo de n bytes. El sufijo b indica un tamaño de n bloques (512 bytes), k indica n kB (1024 bytes) y m indica n MB (1024 kB).

Como cualquier filtro, **split** puede coger un flujo de entrada, lo que ocurre si no se indica ningún archivo, o si hay un guión. Un prefijo define el nombre de los archivos en salida. Aquí tenemos un archivo de 1 GB a cortar en partes de 150 MB. El prefijo es fic. Cada archivo de salida se llama ficaa, ficab, ficac, ficad, y así sucesivamente.

```
$ ls -l granarchivo
-rw-r--r-- 1 seb users 1073741824 mar 12 19:47 granarchivo
$ split -b 150m granarchivo fic
$ ls -l fic*
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficaa
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficab
-rw-r--r-- 1 seb users 157286400 mar 12 20:15 ficac
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficad
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficae
-rw-r--r-- 1 seb users 157286400 mar 12 20:16 ficaf
-rw-r--r-- 1 seb users 130023424 mar 12 20:16 ficag
```

b. Reconstruir

Una línea basta para reconstruir un archivo dividido con la ayuda de las redirecciones:

```
$ cat fic* > newfic
$ ls -l newfic
-rw-r--r-- 1 seb users 1073741824 mar 12 20:47 newfic
```

9. Sustitución de caracteres

a. Lista de caracteres

El comando **tr** permite sustituir unos caracteres con otros y sólo acepta datos que provengan del canal de entrada estándar, no de los archivos.

```
tr [opciones] original destino
```

El original y el destino representan uno o varios caracteres. Se sustituyen los caracteres originales con los de destino en el orden indicado. Los corchetes permiten definir intervalos.

Por ejemplo, para sustituir la o por la e y la i por la a:

```
$ cat lista | tr "oi" "ea"
Predut    ebjete  precie  cantadades
raten     éptaque 30      15
dure      30gaga  100     30
dure      70gaga  150     30
dasce     zap     12      30
dasce     blande  10      30
pantalla  15      150     20
pantalla  17      300     20
pantalla  19      500     20
teclade   105     45      30
teclade   115     55      30
tarjeta   senade  45      30
tarjeta   vadee   145     30
```

Con este comando, puede convertir una cadena en mayúsculas o en minúsculas.

```
$ cat lista | tr "[a-z]" "[A-Z]"
PRODUCTO  OBJETO  PRECIO  CANTIDADES
RATÓN     ÓPTICO  30      15
DURO      30GIGA  100     30
DURO      70GIGA  150     30
DISCO     ZIP     12      30
DISCO     BLANDO  10      30
PANTALLA  15      150     20
PANTALLA  17      300     20
```

```
PANTALLA  19      500     20
TECLADO   105     45      30
TECLADO   115     55      30
TARJETA   SONIDO  45      30
TARJETA   VÍDEO   145     30
```

Eliminar las repeticiones

Sobre todo, `tr` admite dos parámetros, `-s` (squeeze) y `-d` (delete), que permiten suprimir caracteres, duplicados o no. Es perfecto en el caso de separadores múltiples. A continuación damos un ejemplo práctico en el cual se busca aislar la dirección IP de una máquina.

```
$ /sbin/ifconfig eth0
eth0      Vínculo encap:Ethernet  HWaddr 00:13:D3:D7:A4:6C
          inet adr:10.9.238.170  Bcast:10.9.239.255  asque:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15054381 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4991811 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:4157389034 (3964.7 Mb)  TX bytes:374974072 (357.6 Mb)
          Interrupción:22 Dirección básica:0xcc00
```

Sólo le interesa la línea que contiene `inet`:

```
$ /sbin/ifconfig eth0| grep "inet "
          inet dir:10.9.238.170 Bcast:10.9.239.255  Máscara:255.255.252.0
```

Para aislar la dirección IP colocada después de "`inet dir:`" el separador ":" puede parecer interesante, pero en este caso un cut nos devolvería "`10.9.238.170 Bcast`", lo que no conviene. La artimaña consiste en sustituir todos los espacios por un solo ":". El parámetro `-s`sustituye una cadena de n caracteres idénticos por uno solo. Si no se precisa, es el mismo carácter; en caso contrario, se trata de un carácter de sustitución determinado.

```
$ /sbin/ifconfig eth0| grep "inet " | tr -s " " ":"
:inet:dir:10.9.238.170:Bcast:10.9.239.255:Máscara:255.255.252.0
```

Ya sólo falta contar: la dirección IP está en cuarta posición (el primer campo antes del primer ":" está vacío).

```
$ /sbin/ifconfig eth0| grep "inet " | tr -s " " ":" | cut -d: -f4
10.9.238.170
```

b. Tabulaciones y espacios

La mayoría de los editores sustituyen las tabulaciones por espacios. Ahora bien, algunos comandos esperan a obtener tabulaciones como delimitadores de campos (es el caso de cut). Si no puede apañarse con `tr`, tiene a su disposición dos comandos para este caso específico.

El comando **expand** convierte las tabulaciones en espacios. El comando **unexpand** convierte los espacios en tabulaciones. O sea, el archivo lista según el modelo anterior, en el cual se separan las columnas por espacios en lugar de tabulaciones. En el primer caso, el resultado no es el esperado para nada. El comando **cut** intenta sacar el tercer campo de un archivo tabulado. Como no hay tabulaciones, muestra toda la línea.

```
$ cut -f1 lista
Producto     objeto     precio   cantidades
ratón        óptico     30       15
duro         30giga     100      30
duro         70giga     150      30
```

```
disco        zip        12       30
disco        blando     10       30
...
```

El comando **unexpand** con el parámetro `-a` sustituye todas las secuencias de al menos dos espacios por el número necesario de tabulaciones. Esta vez el resultado es correcto.

```
$ unexpand -a lista | cut -f1
Producto
ratón
duro
duro
disco
disco
...
```

10. Visualización de texto

a. En pantalla completa

Nada impide desviar cualquier flujo para visualizarlo en la pantalla o por impresora. Aquí presentamos algunos comandos.

- página por página: **pg**, **more**, **less**
- en bloque: **cat**
- al revés: **tac**
- en dump hexadecimal: **hexdump**
- creación de un banner: **banner**
- formateo para impresión: **pr**
- numerar las líneas: **cat -n** o **nl**

b. El principio de un archivo

Para ver el principio del contenido de un archivo, utilice el comando **head**.

```
head [-c nbcar] [-n nblíneas] [fic1...]
```

El parámetro `-c` permite precisar el número de bytes de encabezamiento que visualizar. Por defecto se visualizan diez líneas. El parámetro `-n` permite indicar el número de líneas que visualizar. Es posible indicar directamente el número de líneas:

```
head [-nblíneas] [fic1...]

$ head -3 lista
Producto objeto   precio  cantidades
ratón    óptico   30      15
duro     30giga   100     30
```

c. Fin y modo de espera de archivo

Para ver las últimas líneas de un archivo, utilice el comando **tail**.

```
tail [+/-valor[b/c]] [-f] [fic1...]
```

Al igual que para head, por defecto se visualizan las diez últimas líneas. El valor `-numlíneas`permite modificar este estado. Use c para indicar un número de caracteres. Una b

indica un número de bloques (512 bytes por bloque).

Finalmente, la opción `-f` deja el archivo abierto. Si se inserta más información en él (por ejemplo, un archivo de registros), se visualizará su contenido en tiempo real en la pantalla hasta que el usuario lo interrumpa de manera voluntaria ([Ctrl] **C**).

```
$ tail -5 lista
pantalla  19      500     20
teclado   105     45      30
teclado   115     55      30
tarjeta   sonido  45      30
tarjeta   vídeo   145     30

$ tail -10c lista
eo        145     30
```

d. Dar formato a una salida

El comando **column** permite dar formato de tabla a la salida de un comando. La opción `-t`determina cuántas columnas se mostrarán en la salida y añade espacios para alinearlas. La opción`-s` permite indicar cuál es el separador.

```
$ column -s: -t /etc/group
root            x  0
daemon          x  1
bin             x  2
sys             x  3
adm             x  4  seb
tty             x  5
disk            x  6
lp              x  7
```

11. Duplicación del canal de salida estándar

En algunos casos, como por ejemplo en la generación de archivos de registros, puede ser necesario colocar el resultado de un comando en un archivo y a la vez filtrar este mismo resultado con otro comando. Para ello, utilice el comando **tee**, que permite duplicar el flujo de datos. Este comando lee el flujo de datos que proviene de otro comando por el canal de entrada, lo escribe en un archivo y restituye este flujo de forma idéntica por el canal de salida. Por defecto, el archivo generado sobrescribe el antiguo si existe.

```
tee [-a] nombre_fic
```

El parámetro `-a` significa append. En este caso, no se sobrescribe el archivo, sino que se inserta la información al final. Por ejemplo, supongamos que quiere obtener en un archivo la lista de los nombres de usuario y que al mismo tiempo se visualice su número en la pantalla.

```
$ cat /etc/passwd | cut -d: -f1 | tee users | wc -l
        65
$ cat users
root
nobody
nobodyV
daemon
bin
uucp
uucpa
auth
cron
lp
tcb
```

```
...
```

12. Comparación de archivos

Los dos comandos que permiten comparar el contenido de dos archivos, o de un archivo y de un flujo, son los comandos **diff** y **cmp**.

a. diff

El comando **diff** indica las modificaciones que hay que aportar a los dos archivos en entrada para que su contenido sea idéntico.

```
diff [-b] [-e] fic1 fic2
```

La opción `-b` permite ignorar los espacios (blank), y la opción `-e` permite generar un script ed (no lo utilizaremos). Este comando devuelve tres tipos de mensajes:

- `APPEND: línea1 a línea3,línea4, ex 5 a 6,8` significa: en la línea 5 de fic1 hay que insertar las líneas 6 a 8 de fic2 para que sus contenidos sean idénticos.
- `DELETE: línea1,línea2 d línea3, ex 7,9 d 6` significa: se deben suprimir las líneas 7 a 9 de fic1, no existen detrás de la línea 6 de fic2.
- `CHANGE: línea1,línea2 c línea3,línea4, ex 8,12 c 9,13` significa: se debe intercambiar las líneas 8 a 12 de fic1 contra las líneas 9 a 13 de fic2.

En cualquier caso, el signo "<" indica las líneas de fic1 concernientes y el signo ">" las líneas de fic2 concernientes.

```
$ cat lista
Producto  objeto  precio  cantidades
ratón     óptico  30      15
duro      30giga  100     30
duro      70giga  150     30
disco     zip     12      30
disco     blando  10      30
pantalla  15      150     20
pantalla  17      300     20
pantalla  19      500     20
teclado   105     45      30
teclado   115     55      30
tarjeta   sonido  45      30
tarjeta   vídeo   145     30

$ cat lista2
Producto  objeto  precio  cantidades
ratón     botones 30      15
duro      30giga  100     30
duro      70giga  150     30
disco     zip     12      30
disco     blando  10      30
pantalla  15      150     20
pantalla  17      300     20
pantalla  19      500     20
pantalla  21      500     20
teclado   105     45      30
teclado   115     55      30
```

El archivo lista es el original. En lista2 se ha modificado la segunda línea, se ha añadido una línea pantalla y se han suprimido las últimas dos líneas.

```
$ diff lista lista2
```

```
2c2
< ratón           óptico  30      15
---
> ratón           botones 30      15
9a10
> pantalla 21     500     20
12,13d12
< tarjeta sonido  45      30
< tarjeta vídeo   145     30
```

- `2c2`: se deben intercambiar las líneas 2 de lista y lista2 (deben concordar o en óptico o en botones).
- `9a10`: después de la línea 9 de lista (pantalla 19), hay que añadir la línea 10 (pantalla 21) de lista2.
- `12,13d12`: se deben suprimir las líneas 12 y 13 de lista (tarjeta de sonido y vídeo), ya que no existen después de la línea 12 de lista2.

b. cmp

El comando **cmp** compara los archivos carácter por carácter. Por defecto, el comando se para en cuanto encuentra la primera diferencia e indica la posición.

```
cmp [-l] [-s] fic1 fic2
```

El parámetro `-l` detalla todas las diferencias en tres columnas. La primera columna representa el número de carácter; la segunda, el valor octal ASCII del carácter correspondiente de fic1, y la tercera, el valor octal ASCII del carácter correspondiente de fic2.

La opción `-s` devuelve únicamente el código de error (no visible), al que se puede acceder por echo $?.

```
$ cmp lista lista2
lista lista2 differ: char 38, line 2
$ cmp -l lista lista2
    38 157 142
    39 160 157
    40 164 165
    41 151 164
    42 161 157
    43 165 156
    44 145 163
   182 143 145
   183 154 143
  ...
```

13. Plazo de espera

El comando **sleep** permite esperar el número de segundos indicados. El script se interrumpe durante este tiempo: el número de segundos y un entero comprendido entre 0 y cuatro mil millones (136 años).

```
$ sleep 10
```

14. Controlar el flujo

El comando **pv**, a menudo desconocido, permite responder a una pregunta frecuente: ¿qué pasa en la tubería mientras que los datos pasan de un proceso a otro? ¿Cuántos datos? ¿Cuánto va a tardar? pv es un monitor de flujo. Se intercala generalmente entre uno y otro comando (como tee), y analiza

el flujo que recibe antes de reenviarlo a su destino. De esta forma, sabe lo que se ha transferido, y puede mostrar una barra de progreso, por ejemplo. Veamos dos ejemplos:

Copiar un archivo

```
[jolivares@slyserver ~]$ pv test > test2
50MiB 0:00:02 [24,1MiB/s] [================>                 ] 52% ETA
```

Duración de la compresión de un archivo

```
pv /boot/vmlinuz-3.14.7-200.fc20.x86_64 | gzip > test.gz
5,26MiB 0:00:00 [6,05MiB/s] [================================>] 100%
```

Los procesos

1. Definición y entorno

Un **proceso** representa un programa en curso de ejecución y, al mismo tiempo, todo su entorno de ejecución (memoria, estado, identificación, propietario, padre...).

Los datos de identificación de un proceso son:

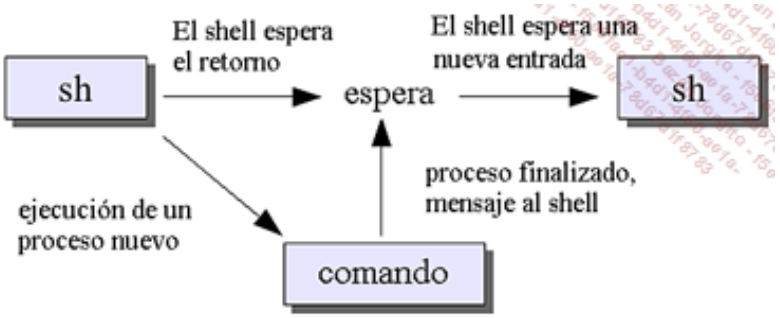
- **Un número de proceso único PID** (*Process ID*): se numera cada proceso Unix con el fin de poder diferenciarlo de los otros. El primer proceso iniciado por el sistema es 1, y se trata de un proceso llamado generalmente init. Se utiliza el PID cuando se trabaja con un proceso. Iniciar 10 veces el mismo programa (mismo nombre) produce 10 PID diferentes.
- **Un número de proceso padre PID** (*Parent Process ID*): cada proceso puede iniciar otros procesos, sus procesos hijos (*child process*). Cada proceso hijo debe contener, entre toda su información, el PID del proceso padre que lo inició. Todos los procesos tienen un PPID salvo el proceso 0, que es un seudoproceso que representa el inicio del sistema (crea el 1 init).
- **Un número de usuario y uno de grupo**: corresponde al UID y al GID de la cuenta de usuario que inicia el proceso. El sistema lo utiliza para determinar, a través de la cuenta, los permisos que el proceso tiene para acceder a los recursos. Los procesos hijos heredan ambas cuentas. En algunos casos (que veremos más adelante) se puede modificar este comportamiento.
- **Duración y prioridad del proceso**: la duración del proceso corresponde al tiempo de ejecución consumido desde la última invocación. En un entorno multitarea, el tiempo de procesador se comparte entre los procesos y no todos tienen la misma prioridad. Los procesos de más alta prioridad son ejecutados primero. Cuando un proceso está inactivo, su prioridad aumenta con el fin de tener la oportunidad de ser ejecutado. Cuando está activo, su prioridad baja con el fin de dejar paso a otro. El planificador de tareas del sistema es el que gestiona las prioridades y los tiempos de ejecución.
- **Directorio de trabajo activo**: tras su inicio, se configura el directorio actual del proceso (desde el cual se inició). Este directorio servirá de base para las rutas relativas.
- **Archivos abiertos**: tabla de los descriptores de archivos abiertos. Por defecto al principio sólo hay tres presentes: 0, 1 y 2 (los canales estándar). Con cada apertura de archivo o de nuevo canal, la tabla se rellena. Al cierre del proceso, se cierran los descriptores (en principio).
- Puede encontrar más información, como el tamaño de la memoria asignada, la fecha de inicio del proceso, el terminal de atribución, el estado del proceso, los UID efectivo y real, así como los GID efectivo y real.

2. Estados de un proceso

Durante su vida (tiempo entre el inicio y la salida) un proceso puede pasar por diversos estados o**process state**:

- ejecución en modo usuario (**user mode**);
- ejecución en modo núcleo (**kernel mode**);
- en espera E/S (**waiting**);
- dormido (**sleeping**);
- listo para la ejecución (**runnable**);
- dormido en el swap (**memoria virtual**);
- nuevo proceso;
- fin de proceso (**zombie**).

3. Ejecución en segundo plano



Según lo visto anteriormente, dado que el entorno de trabajo es multitarea, hay un determinado número de procesos que ya se ejecutan en su máquina sin que usted lo vea. Del mismo modo, el shell que utiliza es en sí mismo un proceso. Cuando inserta un comando, el shell crea un nuevo proceso para ejecutarlo, este proceso se convierte en un proceso hijo del shell. Una vez iniciado, hay que esperar al final de su ejecución para iniciar el siguiente (excepto si se utiliza ";" para encadenar los comandos).

Nada impide al shell esperar el mensaje del proceso terminado para dejar paso: de hecho, una vez iniciado el comando, el shell puede autorizar la inserción de un nuevo comando sin esperar el final de la ejecución del comando anterior. Para ello, basta insertar, después de haber tecleado el comando, el **ampersand "&"**. En este caso, el shell y el comando iniciado funcionarán en paralelo.

```
$ ls -R / > ls.txt 2/dev/null &
[1] 21976
$ [1]    Done                  ls -l -R / > ls.txt 2/dev/null
$ ls
fic1          fic3          lista         ls.txt        dir1          users
fic2          fic4          lista2        mypass        pepito.tar.gz
```

Justo después de la inserción, aparece una cifra. Hay que recordarlo, ya que se trata del PID del nuevo proceso iniciado en segundo plano. Después de otra inserción, un línea Done indica que el tratamiento se da por finalizado. El valor [1] es propio de un shell particular (ksh).

Algunas observaciones en cuanto al uso del inicio en segundo plano:

- El proceso iniciado en segundo plano no debería esperar ser otro shell, dado que se produciría confusión entre este comando y el propio shell.
- El proceso iniciado no debería mostrar resultados en la pantalla porque entrarían en conflicto con los del shell (por ejemplo, aparición de una línea en medio de una inserción).
- Finalmente, cuando se sale del shell, se sale también de todos sus hijos: en este caso, no abandone el shell en mitad de un procesamiento importante.

4. Background, foreground, jobs

Puede retomar el control del shell si ha iniciado un proceso en segundo plano. Puede pararlo de manera temporal tecleando [Ctrl] **Z**:

```
$ sleep 100
<CTRL+Z>
 [1]+  Stopped                 sleep 100
```

Se ha parado el proceso: se ha suspendido su ejecución hasta que vuelva a ponerlo en primer plano con el comando **fg** (*foreground*):

```
$ fg
```

```
sleep 100
```

Cuando ejecuta un comando, obtiene un número entre llaves: es el número de job. Puede obtener una lista de todas las tareas con el comando **jobs**.

```
$ jobs
[1]-  Stopped                 sleep 100
[2]+  Stopped                 sleep 100
```

Los comandos **bg** y **fg** permiten actuar en estos jobs tomando como parámetro su número. Se ejecuta el comando **bg** en un job parado para iniciarlo de nuevo en segundo plano. Se vuelve a iniciar el job 2 en segundo plano:

```
$ bg 2
[2]+ sleep 100 &
$
[2]+  Done                    sleep 100
```

5. Lista de los procesos

El comando **ps** (*process status*) permite obtener información sobre los procesos en curso. Si se ejecuta solo, visualiza únicamente los procesos en curso iniciados por el usuario y desde la consola actual.

```
$ ps
  PID TTY          TIME CMD
 4334 pts/1    00:00:00 bash
 5017 pts/1    00:00:00 ps
```

Para obtener más información, utilice el parámetro `-f`.

```
ps -f
UID        PID  PPID  C STIME TTY          TIME CMD
seb       4334 24449  0 09:46 pts/1    00:00:00 /bin/bash
seb       5024  4334  0 10:51 pts/1    00:00:00 ps -f
```

El parámetro `-e` da información sobre todos los procesos en curso.

```
$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
...
seb      26431     1  0 Mar04 ?        00:00:30 kded [kdeinit]
seb      26436 26322  0 Mar04 ?        00:00:03 kwrapper ksmserver
seb      26438     1  0 Mar04 ?        00:00:00 ksmserver [kdeinit]
seb      26439 26424  0 Mar04 ?        00:00:50 kwin [kdeinit
seb      26441     1  0 Mar04 ?        00:00:28 kdesktop [kdeinit]
seb      26443     1  0 Mar04 ?        00:03:40 kicker [kdeinit]
seb      26453     1  0 Mar04 ?        00:00:00 kerry [kdeinit
seb      26454 26424  0 Mar04 ?        00:00:01 evolution
seb      26465 26424  0 Mar04 ?        00:00:11 kde-window-decorator
seb      26467     1  0 Mar04 ?        00:00:02 gconfd-2 12
seb      26474     1  0 Mar04 ?        00:00:01 knotify [kdeinit]
seb      26485     1  0 Mar04 ?        00:03:06 beagled
...
```

El parámetro `-u` permite precisar una lista de uno o varios usuarios separados por una coma. El parámetro `-g` efectúa lo mismo pero para los grupos, `-t` para los terminales y `-p` para unos PID determinados.

```
$ ps -u root
  PID TTY          TIME CMD
    1 ?        00:00:05 init
    2 ?        00:00:00 kthreadd
    3 ?        00:00:00 migration/0
    4 ?        00:00:09 ksoftirqd/0
    5 ?        00:00:23 events/0
    6 ?        00:00:00 khelper
   25 ?        00:00:00 kblockd/0
   26 ?        00:00:00 kacpid
   27 ?        00:00:00 kacpi_notify
  130 ?        00:00:00 cqueue/0
  131 ?        00:00:00 kseriod
  156 ?        00:00:22 kswapd0
  157 ?        00:00:00 aio/0...
```

Finalmente el parámetro `-l` ofrece más información técnica.

```
$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4704 24449  0  75   0 -  1213 wait   pts/3    00:00:00 bash
```

A continuación presentamos el detalle de algunas columnas.

Columna	Definición
UID	User ID, nombre del usuario.
PID	Process ID, número del proceso.
PPID	Parent Process ID, número del proceso padre.
C	Factor de prioridad, cuanto más grande es el valor, más elevada es la prioridad.
STIME	Hora de inicio del proceso.
TTY	Nombre del terminal desde el cual se ejecutó el proceso.
TIME	Duración de tratamiento del proceso.
CMD	Comando ejecutado.
F	Banderas del proceso (sale del ámbito de este libro).
S	Estado del proceso S (sleeping) R (running) Z (zombie).
PRI	Prioridad del proceso.
NI	Nice, incremento para el scheduler.

6. Parada de un proceso/señales

Es posible detener un proceso en primer plano con la combinación de teclas [Ctrl] **Z**. Cuando un proceso se ejecuta en segundo plano, no lo puede parar cualquier combinación de teclas, al menos que se utilice el gestor de jobs con fg y bg. Puede resultar necesario mandarle señales especiales a ese proceso en segundo plano. Para ello, hay que emplear el comando **kill**. A diferencia de lo que parece indicar su nombre, la función de este comando no es obligatoriamente destruir o terminar un proceso (recalcitrante o no), sino mandar señales a los procesos.

```
kill [-l] -Num_señal PID [PID2...]
```

La **señal** es uno de los medios de comunicación entre los procesos. Cuando se manda una señal a un proceso, éste debe interceptarla y reaccionar en consecuencia. Se pueden ignorar algunas señales, pero otras no. Según los Unix, se dispone de un número más o menos importante de señales. Se numeran y nombran las señales, pero cuidado: si bien los nombres suelen ser comunes de un Unix a

otro, los números no lo son obligatoriamente. La opción **-l** permite obtener la lista de las señales.

Señal	Función
1 (SIGHUP)	El padre manda Hang Up a todos sus hijos cuando termina.
2 (SIGINT)	Interrupción del proceso pedido (tecla [Supr], [Ctrl] **C**).
3 (SIGQUIT)	Ídem SIGINT, pero con generación de un Core Dump (archivo de depuración).
9 (SIGKILL)	Señal que no se puede ignorar, fuerza el proceso a terminar de manera "expeditiva".
15 (SIGTERM)	Señal mandada por defecto por el comando **kill**. Pide al proceso terminar con normalidad.

```
$ sleep 100&
[1] 5187
$ kill 5187
$
[1]+  Completado          sleep 100
$ sleep 100&
[1] 5194
$ kill -9 5194
[1]+  Proceso parado      sleep 100
```

7. nohup

Cuando se sale del shell (exit, [Ctrl] D...) se manda la señal 1 SIGHUP a los hijos para que terminen ellos también. Cuando se inicia un proceso largo en segundo plano y el usuario quiere salir del shell, entonces se para este proceso y habrá que volver a empezarlo. La manera de evitar esto es iniciar el proceso con el comando **nohup**. En este caso, el proceso iniciado ya no reaccionará a la señal SIGHUP, y por lo tanto podrá salir del shell, el comando seguirá su ejecución.

Por defecto, los canales de salida y error estándares se redirigen hacia un archivo **nohup.out**, salvo si se precisa la redirección de manera explícita.

```
$ nohup ls -lR / &
10206
$ Sending output to nohup.out
```

Cuando un proceso hijo finaliza, manda una señal SIGCHLD a su proceso padre. Salvo excepción (un proceso padre desvinculado del hijo), el proceso padre debe obtener tantos SIGCHLD como hijos inició o como SIGHUP emitió. Si el padre termina antes que los hijos, éstos se convertirán en procesos zombis: las señales SIGCHLD son enviadas a... nadie. El proceso hijo ha finalizado correctamente, ha muerto, no consume ningún recurso No se puede matar (ya que está muerto), pero sigue ocupando una entrada en la tabla de los procesos. init lo recupera, y como está siempre en espera, el zombi puede acabar desapareciendo.

8. nice y renice

El comando **nice** permite iniciar un comando con una prioridad más baja para permitir a otros posibles procesos ejecutarse más rápidamente.

```
nice [-valor] comando [argumentos]
```

Un valor positivo causará una bajada de prioridad; uno negativo, el aumento de la prioridad (si está autorizado). El valor debe estar comprendido entre -20 y 20. Cuanto más elevado es el valor, menor es la prioridad.

```
$ nice -10 ls -lR / >lista 2>/dev/null&
10884
$ ps -l
      F S     UID    PID   PPID    C PRI  NI  ADDR   SZ WCHAN    TTY
         TIME CMD
80808001 U N+   75  10884  10822 28.5  54  10     0 848K aa3b3a9c ttyp4
         0:03.32 ls
```

El comando **renice** funciona un poco como nice, pero permite modificar la prioridad en función de un usuario, un grupo o un PID. El proceso debe estar ya ejecutándose.

```
renice [-n prio] [-p] [-g] [-u] ID
```

La prioridad debe estar entre -20 y 20. El usuario estándar sólo puede utilizar los valores entre 0 y 20 que permiten bajar la prioridad. La opción **-p** precisa un PID, **-g** un GID y **-u** un UID.

9. time

El comando **time** mide las duraciones de ejecución de un comando, lo que es ideal para conocer los tiempos de ejecución, y devuelve tres valores:

- **real**: duración total de ejecución del comando;
- **user**: duración del tiempo de CPU necesario para ejecutar el programa;
- **system**: duración del tiempo de CPU necesario para ejecutar los comandos relacionados con el OS (llamadas al sistema dentro de un programa).

El resultado aparece por el canal de error estándar 2. Se puede tener una indicación de la carga de la máquina por el cálculo real / (user+system). Si el resultado es inferior a 10, la máquina dispone de un buen rendimiento; más allá de 20, la carga de la máquina es demasiado pesada (rendimiento reducido).

```
$ time ls -lR /home
...
real    4.8
user    0.2
sys     0.5
```

Más todavía del bash

1. Alias

Un alias es un atajo a un comando y a sus posibles parámetros. Se define con el comando **alias**. Utilizado sin argumentos, lista los alias disponibles.

```
$ alias
alias ..='cd ..'
alias ...='cd ../..'
alias cd..='cd ..'
alias dir='ls -l'
alias l='ls -alF'
alias la='ls -la'
alias ll='ls -l'
alias ls='ls $LS_OPTIONS'
alias ls-l='ls -l'
alias md='mkdir -p'
alias o='less'
alias rd='rmdir'
...
```

Puede crear sus propios alias.

```
$ alias deltree='rm -rf'
```

2. Agrupación de comandos

El encadenamiento de comandos es posible con ";". También es posible agrupar los comandos. Cuando ejecuta los comandos siguientes:

```
$ uname -a ; pwd ; ls -l >resultado.txt &
```

Se ejecuta únicamente el último comando en segundo plano y sólo se redirecciona su resultado en el archivo resultado.txt. Una solución sería:

```
$ uname -a >resultado.txt & ; pwd >>resultado.txt & ; ls -l >>resultado.txt &
[1] 18232
[2] 18238
[3] 18135
```

Es una solución compleja y no funcionará siempre. Además, incluso si se inician los comandos de manera secuencial, se ejecutan todos en paralelo. El primero que finalice será el primero en escribir en el archivo. La solución consiste en utilizar paréntesis.

```
$ (uname -a ; pwd ; ls -l) > resultado.txt &
[1] 18239
$
[1]    Done                  ( uname -a; pwd; ls -l ) > resultado.txt
```

En cualquier caso, se inician todos los comandos colocados entre paréntesis con un subshell, que luego ejecutará los comandos precisados de manera secuencial tal como esté indicado. Así, la redirección concierne al conjunto de los comandos y nada impide lanzar este subshell en segundo plano. De hecho, se distingue claramente un solo PID 18239 durante la ejecución de los comandos.

Otra posibilidad consiste en utilizar llaves {...}. En este caso, no se ejecutará ningún subshell, y si se

ejecuta un comando interno (cd u otro), concierne al shell activo. Se debe colocar la llave de cierre justo después de un ;.

```
$ { uname -a; pwd; ls -l; } > resultado.txt
```

Se puede comprobar la diferencia fácilmente entre las dos sintaxis con exit. El primer ejemplo parece no hacer nada, manteniéndose en el shell hijo. El segundo sale de su shell.

```
$ (exit)
$ { exit; }
```

Cuidado con los paréntesis, en particular en programación. Como se lanza la agrupación dentro de otro proceso, las posibles variables modificadas dentro de la agrupación no serán visibles una vez terminada la ejecución.

3. Relación y ejecución condicional

Además del encadenamiento clásico, se puede relacionar y ejecutar los comandos de manera condicional. La condición de ejecución de un comando es el éxito o no del anterior. Una vez ejecutado, cada comando devuelve un código de retorno, en general 0 si todo ha salido bien, 1 o 2 en caso de error. El shell puede recuperar este valor con la variable $?.

```
$ ls
...
$ echo $?
0
```

Los caracteres **&&** y || permiten efectuar una ejecución condicional.

```
comando1 && comando2
comando1 || comando2
```

Se ejecutará el comando colocado después de **&&** únicamente si el comando anterior ha devuelto 0 (éxito). Sólo se ejecutará el comando colocado después de **||** si el comando anterior ha devuelto algo diferente a 0.

```
$ grep "ratón" lista && echo "Ratón encontrado" || echo "Ratón no encontrado"
ratón  óptico 30      15
Ratón encontrado
$ grep "memoria" lista && echo "Memoria encontrada" || echo "Memoria
imposible de encontrar"
Memoria imposible de encontrar
```

Las variables

Se distinguen tres tipos: usuario, sistema y especiales. El principio consiste en poder asignar un contenido a un nombre de variable, en general una cadena de caracteres o valores numéricos.

1. Nomenclatura

Un nombre de variables obedece a ciertas reglas:

- Se puede componer de letras minúsculas, mayúsculas, cifras, caracteres de subrayado.
- El primer carácter no puede ser una cifra.
- El tamaño de un nombre suele ser ilimitado (pero no hay que pasarse tampoco).
- Las convenciones quieren que las variables de usuario estén en minúsculas para diferenciarlas de las variables de sistema. A elección del usuario.

2. Declaración y asignación

Se declara una variable en cuanto se le asigna un valor. Se efectúa la asignación con el signo **=**, sin espacio antes ni después del signo.

```
var=Hola
```

3. Acceso y visualización

Puede acceder al contenido de una variable colocando el signo **$** delante del nombre de la variable. Cuando el shell encuentra el **$**, intenta interpretar la palabra siguiente como si fuera una variable. Si existe, entonces se sustituye el $nombre_variable por su contenido, o por un texto vacío en el caso contrario. Se habla también de referenciado de variable.

```
$ ruta=/tmp/seb
$ ls $ruta
...
$ cd $ruta
$ pwd
/tmp/seb
$ cd $ruta/dir1
$ pwd
/tmp/seb/dir1
```

Para visualizar la lista de las variables, se utiliza el comando **env**. Muestra las variables de usuario y de sistema, nombre y contenido.

```
$ env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man
KDE_MULTIHEAD=false
SSH_AGENT_PID=26377
HOSTNAME=p64p17bicb3
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
```

```
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
...
```

Una variable puede contener caracteres especiales, particularmente el espacio. El ejemplo siguiente no funciona:

```
$ c=Hola amigos
: not found
$ echo $c
```

Para hacer que funcione, hay que cerrar los caracteres especiales uno por uno, o colocarlos entre comillas o apóstrofes.

```
c=Hola\ Amigos # Solución pesada
c="Hola amigos" # Solución correcta
c='Hola amigos' # Solución correcta
```

La principal diferencia entre las comillas y los apóstrofes es la interpretación de las variables y de las sustituciones. " y ' se cierran mutuamente.

```
$ a=Julio
$ b=César
$ c="$a $b conquistó la Galia"
$ d='$a $b conquistó la Galia'
$ echo $c
Julio César conquistó la Galia
$ echo $d
$a $b conquistó la Galia
$ echo "Linux es genial"
Linux es genial
$ echo 'Linux "demasiado bien"'
Linux "demasiado bien"
```

4. Supresión y protección

Se suprime una variable con el comando **unset**. Puede proteger una variable en modo escritura y contra su supresión con el comando **readonly**. Una variable en modo de sólo lectura, incluso vacía, es exclusiva. No existe ningún medio de sustituirla en escritura y de suprimirla, salvo saliendo del shell.

```
$ a=Julio
$ b=César
$ echo $a $b
Julio César
$ unset b
$ echo $a $b
Julio
$ readonly a
$ a=Nerón
a: is read only
$ unset a
a: is read only
```

5. Export

Por defecto una variable es accesible únicamente desde el shell donde ha sido definida. El siguiente ejemplo declara la variable **a** en el entorno del shell actual y luego trata de visualizar su valor a través de un script lanzado desde este mismo shell. El script no reconoce la variable **a**: no se visualiza nada.

```
$ a=Julio
$ echo 'echo "a=$a"' > ver_a.sh
$ chmod u+x ver_a.sh
$ ./ver_a.sh
a=
```

El comando **export** permite exportar una variable de manera que su contenido sea visible por los scripts y otros subshells. Se pueden modificar las variables exportadas en el script, pero estas modificaciones sólo se aplican al script o al subshell. Esta vez, el primer script puede acceder a la variable **a** exportada. Pero las modificaciones siguen siendo locales en el script. Una vez éste ha terminado, la modificación desaparece.

```
$ export a
$ ./ver_a.sh
a=Julio
$ echo 'a=Nerón ; echo "a=$a"' >> ver_a.sh
$ ./ver_a.sh
a=Julio
a=Nerón
$ echo $a
Julio
```

6. Llaves

Las llaves básicas {} permiten identificar el nombre de una variable. Imaginemos la variable archivo que contiene el nombre de archivo "lista". Quiere copiar lista1 en lista2.

```
$ archivo=lista
$ cp $archivo1 $archivo2
cp: operando archivo que falta
Para saber más, haga: «cp --help».
```

No funciona, ya que no se interpreta $archivo sino $archivo1 y $archivo2, que no existen.

```
$ cp ${archivo}2 ${archivo}1
```

En este caso, esta línea equivale a:

```
$ cp lista2 lista1
```

7. Llaves y sustitución condicional

Las llaves disponen de una sintaxis particular.

```
{variable:Sustitución}
```

Según el valor o la presencia de la variable, es posible sustituir su valor por otro.

Sustitución	Significado
{x:-texto}	Si la variable x está vacía o es inexistente, el texto cogerá su sitio. En caso contrario, es el contenido de la variable el que prevaldrá.
{x:=texto}	Si la variable x está vacía o es inexistente, el texto cogerá su sitio y se convertirá en el valor de la variable.
{x:+texto}	Si la variable x está definida y no vacía, el texto cogerá su sitio. En el caso contrario, una cadena vacía coge su lugar.
{x:?texto}	Si la variable x está vacía o es inexistente, se interrumpe el script

	y se visualiza el mensaje de texto. Si el texto está ausente, se visualiza un mensaje de error estándar.

```
$ echo $nombre

$ echo ${nombre:-Juan}
Juan
$ echo $nombre

$ echo ${nombre:=Juan}
Juan
$ echo $nombre
Juan
$ echo ${nombre:+"Valor definido"}
Valor definido
$ unset nombre
$ echo ${nombre:?Variable ausente o no definida}
nombre: Variable ausente o no definida
$ nombre=Juan
$ echo ${nombre:?Variable ausente o no definida}
Juan
```

8. Variables de sistema

Además de las variables que el usuario puede definir por sí mismo, se inicia el shell con un cierto número de variables predefinidas útiles para ciertos comandos y accesibles por el usuario. Se puede modificar el contenido de estas variables de sistema, pero hay que ser cuidadoso, ya que algunas tienen una incidencia directa sobre el comportamiento del sistema.

Variable	Contenido
HOME	Ruta de acceso del directorio de usuario. Directorio por defecto en caso de uso de CD.
PATH	Lista de directorios, separados por:, donde el shell va a buscar los comandos externos y otros scripts y binarios. La búsqueda se hace en el orden de los directorios insertados.
PS1	Prompt String 1, cadena que representa el prompt estándar visualizado en la pantalla por el shell a la espera de inserción de comando.
PS2	Prompt String 2, cadena que representa un prompt secundario en caso de que se deba completar la inserción.
IFS	Internal Field Separator, lista de los caracteres que separan las palabras en una línea de comandos. Por defecto se trata del espacio, de la tabulación y del salto de línea.
MAIL	Ruta y archivo que contiene los mensajes del usuario.
SHELL	Ruta completa del shell en curso de ejecución.
LANG	Definición del idioma a utilizar, así como del juego de caracteres.
USER	Nombre del usuario en curso.
LOGNAME	Nombre del login utilizado durante el inicio de sesión.
HISTFILE	Nombre del archivo del historial, en general $HOME/.sh_history.
HISTSIZE	Tamaño en número de líneas del historial.
OLDPWD	Ruta de acceso del directorio al que se ha accedido anteriormente.
PS3	Define la línea de comandos para escribir un select.

PWD	Ruta actual.
RANDOM	Genera y contiene un número aleatorio entre 0 y 32767.

9. Variables especiales

Se trata de variables accesibles únicamente para lectura y cuyo contenido suele ser contextual.

Variable	Contenido
$?	Código de retorno del último comando ejecutado.
$$	PID del shell activo.
$!	PID del último proceso iniciado en segundo plano.
$-	Las opciones del shell.

```
$ echo $$
23496
$ grep memoria lista
$ echo $?
1
$ grep ratón lista
ratón  óptico 30      15
$ echo $?
0
$ ls -lR >pepito.txt 2>&1 &
26675
$ echo $!
26675
```

10. Longitud de una cadena

Es posible obtener la longitud de una cadena con el carácter #.

```
$ a=Julio
$ echo "Longitud de $a: ${#a}"
Longitud de Julio: 5
```

11. Tablas y campos

Linux cuenta con dos medios para declarar una tabla: la utilización de los corchetes **[ ]** y la creación global. El primer elemento es 0; el último, 1023. Para acceder al contenido de la tabla, hay que poner la variable Y el elemento entre llaves **{}**.

```
$ Nombre[0]="Julio"
$ Nombre[1]="Román"
$ Nombre[2]="Francisco"
$ echo ${Nombre[1]}
Román
```

o:

```
$ Nombre=(Julio Román Francisco)
$ echo ${nombre[2]}
Francisco
```

Para listar todos los elementos:

```
$ echo ${Nombre[*]}
```

```
Julio Román Francisco
```

Para conocer el número de elementos:

```
$ echo ${#Nombre[*]}
3
```

Si el índice es una variable, no se pone $ delante:

```
$ idx=0
$ echo ${Nombre[idx]}
Julio
```

12. Variables tipadas

Las variables pueden ser de tipo numérico entero (integer) con el comando **typeset -i**. La ventaja es que permite efectuar cálculos y comparaciones sin pasar por expr. El comando **let** o (**(...)**) permite cálculos sobre variables.

Operador	Función
+ - * /	Operaciones sencillas
%	Módulo
< > <= >=	Comparaciones, 1 si verdadero, 0 si falso
== !=	Igual o diferente
&& ||	Comparaciones relacionadas con un operador lógico
& | ^	Lógico binario AND OR XOR

```
$ typeset -i resultado
$ resultado=6*7
$ echo $resultado
42
$ resultado=Error
ksh: Error: bad number
$ resultado=resultado*3
126
$ typeset -i add
$ add=5
$ let resultado=add+5 resultado=resultado*add
$ echo $resultado
50
```

Configuración del bash

1. Archivos de configuración

Se puede lanzar el shell bash en varios modos:

- shell interactiva de conexión (login shell);
- shell interactiva simple;
- shell no interactiva;
- el modo sh;
- etc.

Según su modo de lanzamiento, el shell va a buscar y ejecutar varios scripts y archivos de configuración. Un archivo de configuración es un script de shell, una secuencia de comandos individuales que tiene como meta configurar el entorno del usuario.

a. Shell de conexión

Se lanza el shell de conexión después de la inserción del login y de la contraseña en la consola. Corresponde al que se ha precisado al final de cada línea de /etc/passwd. En este modo, el shell busca ejecutar, en este orden y si están presentes:

- /etc/profile
- ~/.bash_profile
- ~/.bash_login
- ~/.profile

En el momento de desconectarse, intenta ejecutar:

- - ~/.bash_logout

b. Shell simple

El shell interactivo simple corresponde a la ejecución del bash en una ventana (xterm, konsole), una consola o manualmente (teclear bash en una consola). En este caso, sólo se ejecutará el archivo siguiente si existe:

- ~/.bashrc

Observe que en muchas distribuciones Linux, .bashrc es llamado o bien por .bash_profile, o bien por etc/profile y que por lo tanto es la configuración ubicada en .bash_profile la que siempre será invocada.

c. Modo Bourne shell

Cuando se inicia el bash en modo Bourne Shell mediante el comando **sh**, en shell de conexión o no, intenta ejecutar los archivos en este orden:

- /etc/profile
- ~/.profile

d. Modo no interactivo

Se puede iniciar el shell en modo interactivo. Suele ser así cuando ejecuta un script. En este caso, no hay ningún script ejecutado por defecto en el momento del inicio salvo si especifica una variable llamada BASH_ENV que contiene la ruta de un script. En este caso bash carga y ejecuta este archivo antes de comenzar con la ejecución del script o del comando.

2. Comandos set

El shell dispone de una decenas de opciones; la mayoría de ellas se pueden configurar con la ayuda del comando **set**. Las que siguen representan un mero resumen. El - antes de una opción la activa. Un + la desactiva.

Opción	Resultado
-a / -o allexport	Se exportarán todas las variables de manera automática.
-u / -o nounset	Por defecto, el shell trata las variables inexistentes como cadenas vacías. Esta opción produce un error si la variable no existe.
-x / -o xtrace	Muestra todos los comandos a medida que se ejecutan: ideal al principio del script para depuración.
-o vi	Uso de la línea de comandos con la sintaxis de vi.
-o emacs	Uso de la línea de comandos con la sintaxis de emacs.
-C / -o noclobber	Prohíbe las redirecciones de salida si el archivo ya existe.
history	Autoriza la gestión del historial.

El manual del shell le proporcionará todas las opciones posibles.

Programación shell

1. Estructura y ejecución de un script

El shell no es solamente un simple intérprete de comandos, sino que dispone de un verdadero lenguaje de programación; en particular, con gestión de las variables, control del flujo y bucles, operaciones sobre las variables, funciones...

Se agrupan todas las instrucciones y los comandos dentro de un script. Durante su ejecución, cada línea se leerá una por una y se ejecutará. Una línea puede componerse de comandos internos o externos, de comentarios o estar vacía. Es posible diseñar varias instrucciones consecutivas siempre separadas por ; o relacionadas de manera condicional por **&&** o **||**. El **;** es el equivalente de un **salto de línea**.

Por convención, los nombres de los scripts del shell terminan en general (no es obligatorio) por "sh" para el Bourne Shell y el Bourne Again Shell; ".ksh" para el Korn Shell y "csh" para el C Shell.

Para que un script sea ejecutable directamente:

```
$ chmod u+x miscript
```

Para ejecutarlo:

```
$ ./miscript
```

Para evitar el ./:

```
$ PATH=$PATH:.
$ miscript
```

Observe que se coloca el punto en última posición en el PATH. Ponerlo en primera posición puede representar un riesgo para la seguridad: se ha colocado un nuevo comando **ls** modificado en su directorio. Imagine los daños con un comando **passwd**.

Cuando se inicia un script, se crea un nuevo shell hijo que va a ejecutar cada uno de los comandos. Si es un comando interno, el nuevo shell lo ejecuta directamente. Si es un comando externo binario, se creará un nuevo hijo para ejecutarlo. En el caso de un script de shell, se inicia un nuevo shell hijo para leer este nuevo shell línea por línea.

Una línea de comentario siempre empieza con el carácter **#**. Se puede colocar un comentario al final de una línea que ya comporta comandos.

```
# La línea siguiente efectúa un ls
ls # La línea en cuestión
```

La primera línea reviste una importancia particular, ya que permite especificar qué shell va a ejecutar el script:

```
#!/bin/bash
#!/bin/ksh
```

En el primer caso, es un script Bourne Again; en el otro, un script Korn.

2. Argumentos de un script

a. Parámetros de posición

Los parámetros de posición son también variables especiales utilizadas cuando se pasan parámetros a un script.

Variable	Contenido
$0	Nombre del comando (del script).
$1-9	$1,$2,$3... Los nueve primeros parámetros pasados al script.
$#	Número total de parámetros pasados al script.
$*	Lista de todos los parámetros en formato "$1 $2 $3 ...".
$@	Lista de los parámetros en forma de elementos distintos "$1" $2" "$3" ...

```
$ cat param.sh
#!/bin/bash

echo "Nombre: $0"
echo "Número de parámetros: $#"
echo "Parámetros: 1=$1 2=$2 3=$3"
echo "Lista: $*"
echo "Elementos: $@"

$ param.sh juanito jorgito jaimito
Nombre: ./param.sh
Número de parámetros: 3
Parámetros: 1=juanito 2=jorgito 3=jaimito
Lista: juanito jorgito jaimito
Elementos: juanito jorgito jaimito
```

La diferencia entre **$@** y **$*** no salta a la vista. Retome el ejemplo anterior con una pequeña modificación:

```
$ param.sh juanito "jorgito jaimito"
Nombre: ./param.sh
Número de parámetros: 2
Parámetros: 1=juanito 2=jorgito jaimito 3=
Lista: juanito jorgito jaimito
Elementos: juanito jorgito jaimito
```

Esta vez, sólo se pasan dos parámetros. Sin embargo, las listas parecen visualmente idénticas. En realidad, si la primera contiene, en efecto:

```
"juanito jorgito jaimito"
```

La segunda contiene:

```
"juanito" "jorgito jaimito"
```

O sea, dos elementos. En el primer ejemplo, tenía:

```
"juanito" "jorgito" "jaimito"
```

b. Redefinición de los parámetros

Además de listar las variables, la instrucción **set** permite redefinir el contenido de las variables de posición. Con:

```
set valor1 valor2 valor3 ...
```

$1 tomará como contenido valor1; $2, valor2, y así sucesivamente.

```
$ cat param2.sh
#!/bin/bash
echo "Antes:"
echo "Número de parámetros: $#"
echo "Parámetros: 1=$1 2=$2 3=$3 4=$4"
echo "Lista: $*"
set alfred oscar romeo zulu
echo "después set alfred oscar romeo zulu"
echo "Número de parámetros: $#"
echo "Parámetros: 1=$1 2=$2 3=$3 4=$4"
echo "Lista: $*"

$ ./param2.sh juanito jorgito jaimito donald gilito
Antes:
Número de parámetros: 5
Parámetros: 1=juanito 2=jorgito 3=jaimito 4=donald
Lista: juanito jorgito jaimito donald gilito
después set alfred oscar romeo zulu
Número de parámetros: 4
Parámetros: 1=alfred 2=oscar 3=romeo 4=zulu
Lista: alfred oscar romeo zulu
```

c. Reorganización de los parámetros

El comando **shift** permite modificar la posición de los parámetros. Una simple llamada desplaza todos los parámetros una posición, suprimiendo el primero: $2 se convierte en $1, $3 se convierte en $2, y así sucesivamente. El $1 original desaparece. $#, $* y $@ se vuelven a definir en consecuencia.

El comando **shift** seguido de un valor n efectúa un desplazamiento de n elementos. Así, con shift 4 $5 se convierte en $1, $6 se convierte en $2...

```
$ cat param3.sh
#!/bin/bash
set alfred oscar romeo zulu
echo "set alfred oscar romeo zulu"
echo "Número de parámetros: $#"
echo "Parámetros: 1=$1 2=$2 3=$3 4=$4"
echo "Lista: $*"
shift
echo "Después de un shift"
echo "Número de parámetros: $#"
echo "Parámetros: 1=$1 2=$2 3=$3 4=$4"
echo "Lista: $*"

$ ./param3.sh
set alfred oscar romeo zulu
Número de parámetros: 4
Parámetros: 1=alfred 2=oscar 3=romeo 4=zulu
Lista: alfred oscar romeo zulu
Después de un shift
Número de parámetros: 3
Parámetros: 1=oscar 2=romeo 3=zulu 4=
Lista: oscar romeo zulu
```

d. Salida de script

El comando **exit** permite terminar un script. Por defecto, el valor devuelto es 0 (no error), pero se puede especificar cualquier otro valor de 0 a 255. Puede recuperar el valor de salida mediante la variable $?.

```
$ exit 1
```

3. Entorno del proceso

Solamente las variables exportadas son accesibles por un proceso hijo. Si desea visualizar el contexto relacionado con un hijo (en un script por ejemplo), utilice el comando **env**.

```
$ env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man
KDE_MULTIHEAD=false
SSH_AGENT_PID=28012
HOSTNAME=slyserver
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
XDM_MANAGED=/var/run/xdmctl/xdmctl-
:0,maysd,mayfn,sched,rsvd,method=classic
XDG_SESSION_COOKIE=16af07a56781b4689718210047060300-
1211264847.394692-546885666
TMPDIR=/tmp
GTK2_RC_FILES=/etc/gtk-2.0/gtkrc:/usr/share/themes//QtCurve/gtk-
2.0/gtkrc:/home/seb/.gtkrc-2.0-qtengine:/home/seb/.gtkrc-
2.0:/home/seb/.kde/share/config/gtkrc-2.0
KDE_NO_IPV6=1
GTK_RC_FILES=/etc/gtk/gtkrc:/home/seb/.gtkrc:/home/seb/.kde/share/co
nfig/gtkrc
GS_LIB=/home/seb/.fonts
WINDOWID=71303176
MORE=-sl
QTDIR=/usr/lib/qt3
XSESSION_IS_UP=yes
KDE_FULL_SESSION=true
GROFF_NO_SGR=yes
JRE_HOME=/usr/lib/jvm/jre
USER=seb
...
```

El comando **env** permite volver a definir también el contexto del proceso a ejecutar. Puede ser útil cuando el script debe acceder a una variable que no está presente en el entorno del padre, o que no se desea exportar. La sintaxis es:

```
env var1=valor var2=valor ... comando
```

En el caso de bash, env no es indispensable.

```
var1=valor var2=valor ... comando
```

Si la primera opción es el signo -, entonces se debe suprimir todo el entorno existente para sustituirlo por las nuevas variables y los nuevos valores.

```
$ unset a
$ ./ver_a.sh
a=
```

```
$ a=jojo ./ver_a.sh
a=jojo
$ echo a=$a
a=
```

4. Sustitución de comando

El mecanismo de sustitución permite colocar el resultado de comandos simples o complejos en una variable. Debe situar los comandos que va a ejecutar entre dos acentos graves "`".

```
$ mi_unix=`uname`
$ echo ${mi_unix}
Linux
$ maquina=`uname -a | cut -d" " -f5`
echo $maquina
SMP
```

Cuidado: sólo se asigna a la variable el canal de salida estándar. El canal de error estándar sale siempre por pantalla en este caso.

Los acentos graves no son siempre ideales para estos tratamientos de comando. En efecto, si trabaja con varios niveles, debe cerrar los que están dentro de los niveles superiores. Asimismo, el bash permite utilizar en su lugar la sintaxis **$(...)**, que no tiene este problema.

```
$ mi_unix=$(uname)
$ echo ${mi_unix}
Linux
$ maquina=$(uname -a | cut -d" " -f5)
echo
$maquina
SMP
```

No confunda las llaves y los paréntesis. Las primeras aíslan las variables, mientras que los segundos efectúan la sustitución de los comandos.

5. El programa test

El programa **test** permite evaluar escenarios de programación. Se puede recuperar el resultado de una condición con la variable `$?` (código retorno). Si el resultado es 0, entonces se ha cumplido la condición.

a. Pruebas en una cadena

- **test -z "variable"**: cero, retorno OK si la variable está vacía (p. ej.: test -z "$a").
- **test -n "variable"**: no cero, retorno OK si la variable no está vacía (texto cualquiera).
- **test "variable"= cadena**: OK si las dos cadenas son idénticas.
- **test "variable" != cadena**: OK si las dos cadenas son diferentes.

```
$ a=
$ test -z "$a" ; echo $?
0
$ test -n "$a" ; echo $?
1
$ a=Julio
$ test "$a" = Julio ; echo $?
0
```

Tenga cuidado con colocar correctamente las variables que contienen texto entre comillas, porque si la variable está vacía el sistema producirá un bug:

```
$ a=
$ b=pepito
$ [ $a = $b ] && echo "ok"
bash: [: =: unary operator expected
```

Mientras que

```
[ "$a" = "$b" ] && echo "ok"
```

no produce errores.

b. Pruebas sobre los valores numéricos

Se deben convertir las cadenas en valores numéricos para poder evaluarlas. Bash sólo gestiona valores enteros. La sintaxis es:

```
test valor1 opción valor2
```

y las opciones son las siguientes:

Opción	Función
-eq	Equal: igual
-ne	Not Equal: diferente
-lt	Less than: inferior
-gt	Greater than: superior
-le	Less or equal: inferior o igual
-ge	Greater or equal: superior o igual

```
$ a=10
$ b=20
$ test  "$a" -ne "$b" ; echo $?
0
$ test "$a" -ge "$b" ; echo $?
1
$ test "$a" -lt "$b" && echo "$a es inferior a $b"
10 es inferior a 20
```

c. Pruebas sobre todos los archivos

La sintaxis es:

```
test opción nombre_archivo
```

y las opciones son las siguientes:

Opción	Función
-f	Archivo normal.
-d	Un directorio.
-c	Archivo en modo carácter.

-b	Archivo en modo bloque.
-p	Tubería nombrada (named pipe).
-r	Permiso de lectura.
-w	Permiso de escritura.
-x	Permiso de ejecución.
-s	Archivo no vacío (al menos un carácter).
-e	El archivo existe.
-L	El archivo es un vínculo simbólico.
-u	El archivo existe, SUID-Bit activado.
-g	El archivo existe, SGID-Bit activado.

```
$ ls -l
-rw-r--r--   1 seb   users      1392 Ago 14 15:55 dump.log
lrwxrwxrwx   1 seb   users         4 Ago 14 15:21 vínculo_fic1 -> fic1
lrwxrwxrwx   1 seb   users         4 Ago 14 15:21 vínculo_fic2 -> fic2
-rw-r--r--   1 seb   users       234 Ago 16 12:20 lista1
-rw-r--r--   1 seb   users       234 Ago 13 10:06 lista2
-rwxr--r--   1 seb   users       288 Ago 19 09:05 param.sh
-rwxr--r--   1 seb   users       430 Ago 19 09:09 param2.sh
-rwxr--r--   1 seb   users       292 Ago 19 10:57 param3.sh
drwxr-xr-x   2 seb   users      8192 Ago 19 12:09 dir1
-rw-r--r--   1 seb   users      1496 Ago 14 16:12 resultado.txt
-rw-r--r--   1 seb   users      1715 Ago 16 14:55 pepito.txt
-rwxr--r--   1 seb   users        12 Ago 16 12:07 ver_a.sh
$ test -f vínculo_fic1 ; echo $?
1
$ test -x dump.log ; echo $?
1
$ test -d dir1 ; echo $?
0
```

d. Pruebas combinadas por criterios Y, O, NO

Puede efectuar varias pruebas con una sola instrucción. Las opciones de combinación son las mismas que para el comando **find**.

Criterio	Acción
-a	AND, Y lógico
-o	OR, O lógico
!	NOT, NO lógico
(. . .)	agrupación de las combinaciones. Se deben cerrar los paréntesis \(...\).

```
$ prueba -d "dir1" -a -w "dir1" && echo "dir1: directorio, derecho en escritura"
dir1: directorio, derecho en escritura
```

e. Sintaxis ligera

Se puede sustituir la palabra test por los corchetes abiertos y cerrados **[...]**. Hay que colocar un espacio antes y otro después de los corchetes.

```
$ [ "$a" -lt "$b" ] && echo "$a es inferior a $b"
10 es inferior a 20
```

El bash (y el ksh) integra un comando interno que viene a sustituir al programa test. En la práctica, el comando interno es completamente compatible con el comando externo, pero mucho más rápido, ya que no requiere la creación de un nuevo proceso. Para forzar el uso del comando interno, utilice los dobles corchetes **[[...]]**.

```
$ [[ "$a" -lt "$b" ]] && echo "$a es inferior a $b"
10 es inferior a 20
```

6. if ... then ... else

La estructura **if then else fi** es una estructura de control condicional.

```
if <comandos_condición>
then
   <comandos ejecutados si condición cumplida>
else
   <comandos ejecutados si última condición no realizada>
fi
```

También puede especificar **elif**, en realidad un else if. Si no se cumple la última condición, se prueba una nueva.

```
$ cat param4.sh
#!/bin/bash
if [ $# -ne 0 ]
then
        echo "$# parametros en linea de comandos"
else
        echo "Ningun parametro; set alfred oscar romeo zulu"
        set alfred oscar romeo zulu
fi

echo "Numero de parametros: $#"
echo "Parametros: 1=$1 2=$2 3=$3 4=$4"
echo "Lista: $*"

$ ./param4.sh titi pepito
2 parametros en linea de comandos
Numero de parametros: 2
Parametros: 1=pepito 2=titi 3= 4=
Lista: toto titi

$ ./param4.sh
Ningun parametro; set alfred oscar romeo zulu
Numero de parametros: 4
Parametros: 1=alfred 2=oscar 3=romeo 4=zulu
Lista: alfred oscar romeo zulu
```

7. Evaluación múltiple

El comando **case ... esac** permite comprobar el contenido de una variable o de un resultado de manera múltiple.

```
case Valor in
    Modelo1) Comandos ;;
    Modelo2) Comandos ;;
    *) accion_defecto ;;
esac
```

El modelo es o bien un texto simple o bien uno compuesto de caracteres especiales. Cada bloque de comandos relacionados con el modelo se debe terminar con dos puntos y coma. En cuanto se haya

comprobado el modelo, se ejecuta el bloque de comandos correspondiente. El asterisco en la última posición (cadena variable) es la acción por defecto si no se establece ningún criterio; es opcional.

Carácter	Función
*	Cadena variable (incluso vacía)
?	Un solo carácter
[...]	Un intervalo de caracteres
[!...]	Negación del intervalo de caracteres
|	O lógico

```
$ cat case1.sh
#!/bin/bash
if [ $# -ne 0 ]
then
        echo "$# parametros en linea de comandos"
else
        echo "Ningun parametro; set alfred oscar romeo zulu"
        exit 1
fi

case $1 in
        a*)
             echo "Empieza por a"
             ;;
        b*)
             echo "Empieza por b"
             ;;
        fic[123])
             echo "fic1 fic2 o fic3"
             ;;
        *)
             echo "Empieza por cualquier"
             ;;
esac

exit 0
$ ./case1.sh "adios"
Empieza por a
$ ./case1.sh hola
Empieza por b
$ ./case1.sh fic2
fic1 o fic2 o fic3
$ ./case1.sh error
Enpieza por cualquiera
```

8. Introducción de cadena por el usuario

El comando **read** permite al usuario introducir una cadena y colocarla en una o varias variables. Se valida la cadena con [Intro].

```
read var1 [var2 ...]
```

Si se especifican varias variables, la primera palabra irá en var1; la segunda, en var2, y así sucesivamente. Si hay menos variables que palabras, las últimas palabras van en la última variable.

```
$ cat read.sh
#!/bin/bash
echo "Continuar (S/N)? \c"
read respuesta
```

```
echo "repuesta=$respuesta"
case $respuesta in
        S)
                echo "Sí, seguimos"
                ;;
        N)
                echo "No, paramos"
                exit 0
                ;;
        *)
                echo "Error de inserción (S/N)"
                exit 1
                ;;
esac
echo "Usted siguó. Inserte dos palabras o más :\"
read palabra1 palabra2
echo "palabra1=$palabra1\npalabra2=$palabra2"
exit 0
$ ./read.sh
Continuar (S/N)? S
respuesta=S
Sí, seguimos.
Usted siguió. Inserte dos palabras o más: hola amigos
palabra1=hola
palabra2=amigos
```

9. Los bucles

Permiten la repetición de un bloque de comandos ya sea un número limitado de veces o bien de manera condicional. Todos los comandos que se deben ejecutar en un bucle se colocan entre los comandos **do** y **done**.

a. Bucle for

El bucle **for** no se basa en un incremento de valor cualquiera, sino en una lista de valores, archivos...

```
for var in lista
do
    comandos a ejecutar
done
```

La lista representa un determinado número de elementos que se asignarán a var sucesivamente.

Con una variable

```
$ cat for1.sh
#!/bin/bash
for params in $@
do
        echo "$params"
done
$ ./for1.sh test1 test2 test3
test1
test2
test3
```

Lista implícita

Si no especifica ninguna lista a **for**, entonces es la lista de los parámetros la que está implícita. Ningún script anterior hubiera podido parecerse a:

```
for params
do
        echo "$params"
done
```

Con una lista de elementos explícita

Cada elemento colocado después del "in" se utilizará para cada iteración del bucle, uno tras otro.

```
$ cat for2.sh
#!/usr/bin/sh
for params in lista lista2
do
        ls -l $params
done
$ ./for2.sh
-rw-r--r--   1 oracle   system       234 Aug 19 14:09 lista
-rw-r--r--   1 oracle   system       234 Aug 13 10:06 lista2
```

Con criterios de búsqueda sobre nombres de archivos

Si uno o varios elementos de la lista corresponden a un archivo o a un grupo de archivos presentes en la posición actual de la estructura, el bucle for considera el elemento como un nombre de archivo.

```
$ cat for3.sh
#!/bin/bash
for params in *
do
        echo "$params \c"
        type_fic=`ls -ld $params | cut -c1`
        case $type_fic in
                -)      echo "Archivo normal" ;;
                d)      echo "Directorio" ;;
                b)      echo "modo bloque" ;;
                l)      echo "vinculo simbolico" ;;
                c)      echo "modo caracter" ;;
                *)      echo "otro" ;;
        esac

done
$ ./for3.sh
case1.sh Archivo normal
dump.log Archivo normal
for1.sh Archivo normal
for2.sh Archivo normal
for3.sh Archivo normal
vinculo_fic1 vinculo simbolico
vinculo_fic2 vinculo simbolico
lista Archivo normal
lista1 Archivo normal
lista2 Archivo normal
param.sh Archivo normal
param2.sh Archivo normal
param3.sh Archivo normal
param4.sh Archivo normal
read.sh Archivo normal
dir1 Directorio
resultado.txt Archivo normal
pepito.txt Archivo normal
ver_a.sh Archivo normal
```

Con un intervalo de valores

Existen dos métodos para contar de 1 a n con un bucle for. El primero consiste en utilizar una sustitución de comando con el comando **seq**. Su sintaxis básica requiere un parámetro numérico y cuenta de 1 al valor de este parámetro. El manual le mostrará que se puede iniciar en cualquier valor e incrementarse con cualquier valor.

```
$ seq 5
1
2
3
4
5
```

Usándolo con el bucle for, obtenemos este resultado:

```
$ for i in $(seq 5); do echo $i; done
1
2
3
4
5
```

El segundo método consiste en utilizar una sintaxis parecida a la del lenguaje C:

```
$ for ((a=1 ; a<=5 ; a++)); do echo $a; done
1
2
3
4
5
```

Con una sustitución de comando

Se puede colocar cualquier comando que produzca una lista de valores a continuación del "in" con la ayuda de una sustitución de comando. El bucle for tomará el resultado de este comando como lista de elementos con la que iterar.

```
$ cat for4.sh
#!/bin/bash
echo "Lista de los usuarios en /etc/passwd"
for params in `cat /etc/passwd | cut -d: -f1`
do
        echo "$params "
done
$ ./for4.sh
Lista de los usuarios en /etc/passwd
root
nobody
nobodyV
daemon
bin
uucp
uucpa
auth
cron
lp
tcb
adm
ris
```

```
carthic
ftp
stu
...
```

b. Bucle while

El comando **while** permite un bucle condicional "mientras". Mientras se cumpla la condición, se ejecuta el bloque de comandos. Si la condición ya no es válida, se sale del bucle.

```
while condición
do
    comandos
done
```

o:

```
while
bloque de instrucciones que forman la condición
do
    comandos
done
```

Por ejemplo:

```
$ cat while1.sh
#!/bin/bash
while
        echo "¿Cadena? \c"
        read nombre
        [ -z "$nombre" ]
do
        echo "ERROR: no inserción"
done
echo "Usted insertó: $nombre"
```

Lectura de un archivo línea por línea

```
#!/bin/bash
cat pepito.txt | while read line
do
        echo "$line"
done
```

o:

```
#!/bin/bash
while read line
do
           echo "$line"
done < pepito.txt
```

Hay una enorme diferencia entre los dos versiones. En la primera, observe la presencia de la tubería (pipe): se ejecuta el bucle en un segundo proceso. ¡Por lo tanto, cualquier variable modificada en el interior de este bucle pierde su valor a la salida!

c. Bucle until

El comando **until** permite un bucle condicional "hasta". En cuanto se haya cumplido la condición, se sale del bucle.

```
until condición
do
     comandos
done
```

o:

```
until
bloque de instrucciones que forman la condición
do
     comandos
done
```

d. true y false

El comando **true** no hace más que volver a enviar 0. El comando **false** siempre devuelve 1. De esta manera, es posible realizar bucles sin fin. La única manera de salir de estos bucles es con un exit o un break.

Por convención, cualquier programa que no devuelve error devuelve 0, mientras que cualquier programa que devuelve un error, o un resultado que se debe interpretar, devuelve algo distinto a 0. Es lo contrario en lógica booleana.

```
while true
do
     comandos
     exit / break
done
```

e. break y continue

El comando **break** permite interrumpir un bucle. En este caso, el script continúa su ejecución después del comando **done**. Puede coger un argumento numérico que indique el número de bucles que saltar, en el caso de los bucles anidados (rápidamente ilegible).

```
while true
do
     echo "¿Cadena? \c"
     read a
     if [ -z "$a" ]
     then
         break
     fi
done
```

El comando **continue** permite saltarse líneas de una iteración e ir directamente a la siguiente. Puede coger un argumento numérico, que indica el número de bucles que se deben iniciar de nuevo (se remontan n bucles). El script se ejecuta con el comando **do**.

f. Bucle select

El comando **select** permite crear menús simples, con selección por número. La inserción se efectúa por el teclado con el prompt de la variable PS3. Si el valor insertado no es correcto, se efectúa un bucle y se visualiza de nuevo el menú. Para salir de un select, hay que utilizar un **break**.

```
select variable in lista_contenido
```

```
do
      tratamiento
done
```

Si no se especifica `in lista_contenido`, se utilizarán y visualizarán los parámetros de posición.

```
$ cat select.sh
#!/bin/bash
PS3="Su elección:"
echo "¿Qué dato?"
select respuesta in Julio Román Francisco sale
do
         if [[ "$respuesta" = "sale" ]]
         then
                 break
         fi
         echo "Usted eligió $respuesta"
done
echo "Adiós."
exit 0

$ ./select.sh
¿Qué dato?
1) Julio
2) Román
3) Francisco
4) sale
Su elección :1
Usted eligió Julio
Su elección :2
Usted eligió Román
Su elección :3
Usted eligió Francisco
Su elección :4
Adiós.
```

10. Las funciones

Las funciones son trozos de scripts con nombre, directamente llamados por su nombre, que pueden aceptar parámetros y devolver valores. Los nombres de funciones siguen las mismas reglas que las variables, excepto que no se pueden exportar.

```
nombre_funcion ()
{
      comandos
      return
}
```

Las funciones pueden escribirse o bien en el script actual, o bien en otro archivo que puede incluirse en el entorno. Para ello, teclee:

```
. nombrefic
```

El punto seguido de un nombre de archivo carga su contenido (funciones y variables) en el contexto actual.

El comando **return** permite asignar un valor de vuelta a una función. Bajo ningún concepto se debe utilizar el comando **exit** para salir de una función porque esta instrución interrumpe también el script invocante.

```
$ cat funcion
ll ()
```

```
{
        ls -l $@
}
 li ()
{
        ls -i $@
}
$ . funcion
$ li
  252 case1.sh      326 for4.sh       187 param.sh      897 resultado.txt
  568 dump.log      409 vinculo_fic1  272 param2.sh     991 pepito.txt
  286 funcion       634 vinculo_fic2  260 param3.sh     716 ver_a.sh
  235 for1.sh      1020 lista          42 param4.sh    1008 while1.sh
  909 for2.sh       667 lista1        304 read.sh
  789 for3.sh      1006 lista2        481 dir1
```

11. Cálculos y expresiones

a. expr

El comando **expr** permite efectuar cálculos sobre valores numéricos, comparaciones, así como la búsqueda en cadenas de texto.

Operador	Función
+	Suma.
-	Sustracción.
*	Multiplicación. Como el shell reconoce la estrella como comodín, hay que cerrarla con una contrabarra: *.
/	División.
%	Módulo.
!=	Diferente. Visualiza 1 si diferente, 0 en caso contrario.
=	Igual. Visualiza 1 si igual, 0 en caso contrario.
<	Inferior. Visualiza 1 si inferior, 0 en caso contrario.
>	Superior. Visualiza 1 si superior, 0 en caso contrario.
<=	Inferior o igual. Visualiza 1 si inferior, 0 en caso contrario.
>=	Superior o igual. Visualiza 1 si superior, 0 en caso contrario.
:	Búsqueda en una cadena. P. ej.: expr Julio: J* devuelve 1, ya que Julio empieza por J. Sintaxis particular: expr "Julio": ".*" devuelve la longitud de la cadena.

```
$ expr 7 + 3
10
$ expr 7 \* 3
21
$ a=$(expr 13 - 10)
$ echo $a
3
$ cat expr1.sh
#!/bin/bash
cumul=0
contador=0
nb_bucles=10
while [ "$contador" -le "$nb_bucles" ]
do
```

```
        cumul=$(expr $cumul + $contador)
        echo "$cumul=$cumul, bucle=$contador"
        contador=$(expr $contador + 1)
done
$ ./expr1.sh
cumul=0, bucle=0
cumul=1, bucle=1
cumul=3, bucle=2
cumul=6, bucle=3
cumul=10, bucle=4
cumul=15, bucle=5
cumul=21, bucle=6
cumul=28, bucle=7
cumul=36, bucle=8
cumul=45, bucle=9
cumul=55, bucle=10
$ expr "Julio César" : ".*"
11
```

b. Cálculos con bash

El bash propone una forma sencilla de cálculos sobre los enteros colocando la operación entre**$((...))**:

```
$ a=1
$ a=$((a+1))
$ echo $a
2
$ b=2
$ a=$((a*b))
$ echo $a
4
```

No necesita especificar los $ de los nombres de las variables entre los dobles paréntesis.

12. Punteros

Aquí tiene un ejemplo:

```
$ a=Julio
$ b=a
$ echo $b
a
```

¿Cómo visualizar el valor de a y no su nombre? Utilizando el comando **eval**. Este comando, situado al principio de la línea, intenta interpretar el valor de una variable precedida de dos "$" como si fuera otra variable.

```
$ eval echo \$$b
Julio

$ cat eval.sh
cpt=1
for i in a b c
do
    eval $i=$cpt
    cpt=$((cpt+1))
done
echo $a $b $c

$ ./eval.sh
```

```
1 2 3
```

13. Tratamiento de señales

El comando **trap** permite modificar el comportamiento del script en el momento de la recepción de una señal.

Comando	Reacción
trap " señales	Ignora las señales. trap " 2 3 ignora las señales 2 y 3.
trap 'comandos' señales	Para cada señal recibida, ejecuta los comandos indicados.
trap señales	Restaura las acciones por defecto para estas señales.

En el ejemplo siguiente, antes de haber mostrado su PID y esperar que se pulse la tecla [Entrar],**trap** impide la ejecución de [Ctrl] **C** (SIGINT) e intercepta la señal SIGTERM:

```
$ cat trap.sh
!#/bin/bash

salir()
{
        echo "Señal 15 recibido"
        exit 0
}

trap '' 2
trap exit 15
echo "Yo soy $$"

read
while true
do
        echo "Ctrl+C imposible!"
done

$ ./trap.sh

Yo soy 12456
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
...
```

Desde otra consola:

```
$ kill -2 12456 # ningun efecto
$ kill -15 12456 # SIGTERM
En la primera consola:
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Ctrl+C imposible!
Señal 15 recibida
```

14. Comando ":"

El comando **":"** suele ser totalmente desconocido para los usuarios de Unix. Siempre devuelve el valor 0 (éxito). Por lo tanto, se puede utilizar para sustituir al comando **true** en un bucle, por ejemplo:

```
while:
do
    ...
done
```

Este comando, colocado al principio de línea y seguido de otro comando, trata el comando y sus argumentos, pero no hace nada y siempre devuelve 0. Puede ser útil para probar variables.

```
$ : ls
$ : ${X:?"Error"}
X : Error
```

SQL

1. Presentación

El SQL, *Structured Query Language*, es un lenguaje estandarizado ISO de consulta y tratamiento de bases de datos relacionales. Se pueden descomponer sus posibilidades en cuatro funciones:

- El lenguaje de **definición de datos**: creación, modificación y supresión de las tablas, índice, coacciones, etc.
- El lenguaje de **tratamiento de datos**: seguramente el más famoso. Concierne a las peticiones "clásicas": añadido, supresión, modificación y selección de datos.
- El lenguaje de **control de datos**: instalación y gestión de los privilegios de los usuarios de la base de datos.
- El lenguaje de **control de las transacciones**: gestión en particular de los "commit" y de las posibles vueltas atrás, procedimientos, etc.

Puede ser útil para un administrador saber crear peticiones en una base de datos: algunas herramientas se sirven de bases SQL para almacenar sus datos. Pero incluso en el caso contrario, como se utiliza mucho y es muy práctico, resulta útil conocer SQL. Esta iniciación le muestra cómo efectuar las cuatro peticiones básicas: selección, añadido, modificación y supresión de datos.

El modelo básico de datos utilizado para esta presentación se halla en el anexo de este libro.

2. Peticiones de selección

a. Select

Las selecciones, con la instrucción SELECT, son las principales peticiones utilizadas en SQL, que permiten extraer datos de una o varias tablas según determinados criterios.

```
SELECT apellido_campo1, apellido_campo2...
FROM table;
```

Se devuelve el resultado en forma de tabla cuyos encabezamientos son los nombres de los campos seleccionados. Es posible renombrarlos con AS. Por ejemplo:

```
SELECT apellido, nombre, id AS identificador FROM t_usuarios;
```

La utilización del asterisco "*" como campo permite seleccionar todos los campos de la tabla. La petición siguiente lista todo el contenido de la tabla:

```
SELECT * FROM t_usuarios;
```

b. Distinct

Si la petición devuelve líneas idénticas, puede suprimir los duplicados con `DISTINCT`. En el caso siguiente, es lógico pensar que varios usuarios tienen el mismo nombre. La causa DISTINCT va a suprimir en salida los duplicados (o triplicados, etc.).

```
SELECT DISTINCT nombre FROM t_usuarios;
```

c. Where

La cláusula WHERE especifica las condiciones de selección de las líneas.

```
SELECT apellido_campo1, apellido_campo2...
FROM table
WHERE condición;
```

Es posible aplicar una condición a cualquier campo de la tabla, incluyendo aquellos que no están presentes en la selección de campos. Teniendo en cuenta el tipo de datos del campo o de los campos afectados por la condición, puede utilizar distintos operadores: =, >=, <=, >, <, <> (diferente) y la negación con el signo de exclamación: != (diferente), !> (no superior), !< (no inferior). Se pueden relacionar los criterios de condiciones con los operadores lógicos AND, OR y NOT. Si el criterio es un texto, se pone entre comillas simples: 'titi'.

```
SELECT nombre FROM t_usuarios WHERE apellido='SANCHEZ';
```

Atributos de Where

Puede utilizar los atributos LIKE, BETWEEN o IN en una cláusula WHERE.

- LIKE compara el valor de un campo con un valor de texto especificado con posibles caracteres de sustitución: "%" para una cadena cualquiera y "_" para un carácter cualquiera.
- BETWEEN especifica un intervalo de datos.
- IN propone una lista de elementos.

Esta petición extrae todos los registros cuyos id están incluidos entre 0 y 100.

```
SELECT * FROM t_usuarios WHERE id BETWEEN 0 AND 1000;
```

Ésta extrae todos los registros en los que el nombre corresponde a un nombre compuesto "Juan": Juan Jacobo, Juan Carlos, Juan María, etc.

```
SELECT * FROM t_usuarios WHERE nombre LIKE 'Juan %';
```

Finalmente, ésta última extrae los usuarios que viven en Madrid, Barcelona, Valencia y Sevilla.

```
SELECT * FROM t_usuarios WHERE ciudad IN ('Madrid', 'Barcelona',
'Valencia', 'Sevilla');
```

3. Las expresiones y las funciones

Puede utilizar expresiones, en particular las aritméticas, dentro de una instrucción SELECT, tanto en la selección de los campos como en la cláusula WHERE. La petición siguiente calcula el precio NETO de un producto considerando que el campo iva contiene el porcentaje de IVA aplicado al producto.

```
SELECT precio+(precio*iva) AS precio_net FROM t_productos
WHERE id_producto='1245';
```

Si los cambos son de texto, el símbolo "+" permite hacer concatenaciones:

```
SELECT apellido+' '+nombre FROM t_usuarios;
```

Las funciones se aplican como las expresiones en los campos seleccionados o en la cláusula WHERE. Son de tres tipos principales: matemáticas y estadísticas, fechas y cadenas de caracteres. Le presentamos unos ejemplos.

Esta consulta cuenta el número de líneas en una tabla:

```
SELECT count(*) FROM t_usuarios;
```

Ésta extrae los precios mínimo, máximo y medio del conjunto de la tabla de los productos:

```
SELECT min(precio), max(precio), avg(precio) FROM t_productos;
```

Esta última extrae el importe total de las existencias de productos:

```
SELECT sum(precio*ctd) from t_productos;
```

Las funciones de fecha se aplican sobre los campos de tipo fecha, salvo `current_date`, que devuelve la fecha del día:

```
SELECT current_date;
```

La petición siguiente extrae todos los usuarios que se han inscrito en 2008:

```
SELECT * from t_usuarios WHERE month(f_inscripción)='2008';
```

Finalmente, aquí tiene un ejemplo de petición que extrae todos los usuarios cuyo nombre empieza por un 'RI' al recuperar los dos primeros caracteres del nombre con la función LEFT:

```
SELECT * FROM t_usuarios WHERE left(nombre,2) = 'RI';
```

4. La cláusula ORDER BY

La cláusula `ORDER BY` permite ordenar los resultados. Es posible indicar el tipo de ordenación, creciente con `ASC`, decreciente con `DESC`. La petición siguiente ordena los productos de más barato a más caro teniendo en cuenta el IVA:

```
SELECT precio+(precio*iva) AS precio_net FROM t_productos ORDER BY
precio_net ASC;
```

5. La cláusula GROUP BY

La cláusula `GROUP BY` agrupa los resultados por campos de valores idénticos. La petición siguiente visualiza el id de proveedor y número total de productos en reserva por id de proveedor:

```
SELECT id_proveedor, sum(ctd) FROM t_productos GROUP BY id_proveedor;
```

La cláusula `HAVING` se utiliza con GROUP BY y permite reducir la selección con un criterio en el campo o los campos de agrupación. Por ejemplo, en la petición de más arriba, puede agrupar únicamente en un solo proveedor dado:

```
SELECT id_proveedor, sum(ctd) FROM t_productos GROUP BY
id_proveedor HAVING id_proveedor='1';
```

6. Las combinaciones

Hasta ahora las consultas afectaban a una única tabla, pero con SQL es posible recuperar registros de varias tablas al mismo tiempo. Para ello las tablas implicadas deben compartir campos que las puedan relacionar. La combinación de tablas se basa en la relación de varias tablas a través de uno (o varios) campos (atributos) de cada tabla. No hay límite al número de tablas ni al de combinaciones.

```
SELECT tabla1.campo1, tabla2.campo1
```

```
FROM tabla1, tabla2
WHERE tabla1.campocomun=tabla2.campocomun;
```

A continuación presentamos una consulta que extrae el nombre de cada proveedor para cada producto:

```
SELECT DISTINCT t_proveedores.nombre
FROM t_proveedores, t_productos
WHERE t_proveedores.id_proveedor=t_proveedores.id_proveedor;
```

Se puede utilizar el AS para renombrar las tablas en la salida de la consulta. Si el nombre del campo es único en todas las tablas, entonces no hace falta especificar la tabla. Se puede transformar la consulta de la manera siguiente:

```
SELECT DISTINCT nombre
FROM t_proveedores as t1, t_productos as t2
WHERE t2.id_proveedor=t1.id_proveedor
```

Se pueden efectuar combinaciones con más de dos tablas. Así, para obtener la lista de los proveedores por clientes, necesitará consultar la tabla de los clientes llamada "t_usuarios", la tabla de los pedidos, la tabla de los productos y la tabla de los proveedores:

```
SELECT DISTINCT t_proveedores.nombre
FROM t_usuarios, t_pedidos, t_productos
WHERE t_usuarios.id=t_pedidos.id
AND t_pedidos.id_producto=t_productos.id_producto
AND t_productos.id_proveedor=t_proveedores.id_proveedor
AND t_usuarios.id='1'
```

Es posible combinar una tabla consigo misma. El truco consiste en duplicar temporalmente la tabla después de FROM, pero con nombres diferentes especificados con AS. El ejemplo siguiente extrae los apellidos y nombres de alias del usuario 2:

```
SELECT B.apellido, B.nombre
FROM t_usuarios AS A, t_usuarios AS B
WHERE B.id = A.id_alias
AND A.id = '2';
```

7. Subconsultas

Es posible hacer selecciones anidadas: que la devolución de una primera selección sirve de criterio para una segunda selección. Por ejemplo, veamos cómo extraer los apellidos y nombres de todos los alias:

```
SELECT id
FROM t_usuarios
WHERE id IN (select id_alias from t_usuarios WHERE id_alias!='')
```

Si la subconsulta devuelve un único valor, puede usar un igual para pasarle el valor a la consulta superior. En caso de varios valores devueltos, utilice IN.

8. Las inserciones

Inserte registros en una tabla SQL con el comando `INSERT`.

```
INSERT INTO table (campo1, campo2, etc.)
VALUES ('valor1', 'valor2'...);
```

Por ejemplo, veamos cómo insertar un nuevo cliente en la tabla t_usuarios:

```
INSERT INTO t_usuarios (id, apellido, nombre, ciudad, id_alias)
VALUES ('3', 'Aguilera', 'Julio', 'Olmo', NULL);
```

9. Actualizaciones

La instrucción `UPDATE` actualiza uno o varios registros de una tabla. La cláusula `WHERE` es opcional. En el ejemplo, se actualizan todos los registros.

```
UPDATE tabla SET campo1='valor', campo2='valor'
[WHERE campon='valor']
```

Veamos cómo aumentar todos los precios un 5 %:

```
UPDATE t_productos SET precio=precio*1.05;
```

10. Eliminación

La instrucción `DELETE` suprime uno o varios registros de una o varias tablas. Cuidado, según el modelo básico de datos, con las restricciones de integridad: en algunos casos, suprimir una referencia puede suprimir n registros en cascada.

```
DELETE FROM tabla
WHERE campo1='valor';
```

Veamos cómo suprimir el usuario 3 creado más arriba:

```
DELETE from t_usuarios WHERE id=3;
```

Representación de los discos

Nota previa: las unidades de medida de almacenamiento usadas en este capítulo y en todo el libro usan la representación tradicional, según la regla 1KB = 1024 bytes (2^{10}), a no ser que se indique lo contrario.

1. Nomenclatura

Este apartado realiza un repaso a los puntos ya vistos en el capítulo Presentación de Linux. En función del tipo de controlador e interfaz en los cuales se conectan los discos, Linux da diferentes nombres a los archivos especiales que representan discos duros.

Cada disco y cada partición está representado por un archivo especial de tipo bloque.

a. IDE

Los discos con controladores IDE (también llamados PATA, *Parallel Ata* o ATAPI) se llaman hdX:

- hda: IDE0, Master
- hdb: IDE0, Slave
- hdc: IDE1, Master
- hdd: IDE1, Slave
- etc.

Contrariamente a lo que se cree, no hay límite al número de controladores IDE, más allá del número de los puertos de extensión de la máquina (slots PCI). Hay muchas tarjetas adicionales y placas base que disponen de cuatro, seis, ocho conectores. En estos casos, los archivos especiales reciben los nombres hde, hdf, hdg, etc.

Linux entiende que los lectores de CD-Rom, DVD y grabadores son discos IDE y respeta la nomenclatura citada.

Los últimos núcleos de Linux utilizan por defecto un API llamado libata para acceder al conjunto de los discos IDE, SCSI, USB, Firewire, etc. En este caso (puede comprobarlo dirigiéndose a las notas de versión de la distribución), la nomenclatura sigue la de los discos SCSI, que tratamos en el punto siguiente.

La representación de los discos con nomenclatura hdx se ha vuelto cada vez más rara. Sin embargo tenga cuidado. Si actualiza su antigua distribución que mantiene ese formato, el paso a una nueva versión más reciente probablemente cambiará el nombre a sdx, la nomenclatura SCSI. Así que puede experimentar problemas si olvida modificar el archivo /etc/fstab o la configuración de grub.

b. SCSI, SATA, USB, FIREWIRE, etc.

Los discos con controladores SCSI, SCA, SAS, FiberChannel, USB, Firewire (y probablemente otras interfaces exóticas, como los lectores ZIP en puerto paralelo) se llaman sdX. La enumeración de los discos sigue el orden de detección de las tarjetas SCSI y de los adaptadores (hosts) asociados, más la adición o supresión manual de otras interfaces de discos duros mediante hotplug o udev.

- sda: primer disco SCSI
- sdb: segundo disco SCSI
- sdc: tercer disco SCSI

- etc.

La norma SCSI marca una diferencia entre los diversos soportes. Así, los lectores de CD-Rom, DVD, HD-DVD, BlueRay y los grabadores asociados no llevan el mismo nombre. Los lectores y grabadores están en srX (sr0, sr1, etc.). También puede encontrar scd0, scd1, etc. Pero suelen ser vínculos simbólicos hacia sr0, sr1, etc.

El comando **lsscsi** permite enumerar los periféricos SCSI.

```
$ lsscsi
[4:0:0:0]    disk    ATA      ST380011A        8.01  /dev/sda
[5:0:0:0]    cd/dvd  LITE-ON  COMBO SOHC-4836V S9C1  /dev/sr0
[31:0:0:0]   disk    USB2.0   Mobile Disk      1.00  /dev/sdb
```

2. Casos especiales

a. Controladores específicos

Algunos controladores no siguen esta nomenclatura. Por ejemplo, es el caso de algunos controladores RAID físicos. Hay que verlo caso por caso. Un controlador Smart Array en un servidor HP coloca sus archivos de periféricos en /dev/cciss con los nombres cXdYpZ, donde X es el slot, Y el disco y Z la partición...

b. Virtualización

La representación de discos de sistemas invitados (*guests*) virtualizados depende del tipo de controlador simulado. La mayoría son de tipo IDE o SCSI, y en ambos casos muy a menudo con libata son vistos como SCSI. Sin embargo, algunos sistemas, como por ejemplo KVM que ofrece paravirtualización, disponen de un controlador específico que presenta los discos con el nombre vd_x (virtual disk x o xvd_x):

- vda: primer disco virtualizado, o vxda.
- vdb: segundo disco virtualizado, o vxbd.
- etc.

c. SAN, iSCSI, multipathing

Los discos conectados a través de una SAN (*Storage Area Network*, generalmente con fibra óptica) o mediante iSCSI se ven como discos SCSI y conservan esta nomenclatura. Sin embargo, los sistemas de gestión de rutas múltiples (*multipathing*) se ubican por debajo, proporcionando otros nombres. Powerpath llamará a los discos emcpowerx (emcpowera, emcpowerb, etc.) mientras que el sistema por defecto de Linux llamado multipath los llamará mpathx (mpath0, mpath1, etc.) o de cualquier otro modo elegido por el administrador.

Operaciones de bajo nivel

1. Información

El comando **hdparm** permite efectuar un gran número de operaciones directamente en los discos duros gestionados por la librería libata, o sea todos los discos SATA, ATA (IDE) y SAS. El comando**sdparm** puede hacer más o menos lo mismo para los discos SCSI. Observe que, a pesar de que los nombres de periféricos de la libata sean idénticos a los del SCSI, es más que probable que muchas opciones de configuración de hdparm no funcionen en discos SCSI. Lo mismo vale para sdparm con los discos SATA o IDE. Los ejemplos que damos a continuación se basan en hdparm.

Para obtener información completa relativa a un disco, utilice los parámetros `-i` o `-I`. El primero recupera la información, desde el núcleo, que se obtiene en el momento del arranque. El segundo interroga directamente al disco. Es preferible `-I` porque da una información muy detallada.

```
# hdparm -I /dev/sda

/dev/sda:

ATA device, with non-removable media
        Model Number:       ST380011A
        Serial Number:      5JVTH798
        Firmware Revision:  8.01
Standards:
        Used: ATA/ATAPI-6 T13 1410D revision 2
        Supported: 6 5 4
Configuration:
        Logical         max     current
        cylinders       16383   16383
        heads           16      16
        sectors/track   63      63
        --
        CHS current addressable sectors:   16514064
        LBA    user addressable sectors:  156301488
        LBA48  user addressable sectors:  156301488
        device size with M = 1024*1024:       76319 MBytes
        device size with M = 1000*1000:       80026 MBytes (80 GB)
Capabilities:
        LBA, IORDY(can be disabled)
        Standby timer values: spec'd by Standard, no device specific
minimum
        R/W multiple sector transfer: Max = 16  Current = 16
        Recommended acoustic management value: 128, current value: 0
        DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5
             Cycle time: min=120ns recommended=120ns
        PIO: pio0 pio1 pio2 pio3 pio4
             Cycle time: no flow control=240ns  IORDY flow control=120ns
Commands/features:
        Enabled Supported:
            *    SMART feature set
                 Security Mode feature set
            *    Power Management feature set
            *    Write cache
            *    Look-ahead
            *    Host Protected Area feature set
            *    WRITE_BUFFER command
            *    READ_BUFFER command
            *    DOWNLOAD_MICROCODE
                 SET_MAX security extension
            *    48-bit Address feature set
            *    Device Configuration Overlay feature set
```

```
            *    Mandatory FLUSH_CACHE
            *    FLUSH_CACHE_EXT
            *    SMART error logging
            *    SMART self-test
            *    General Purpose Logging feature set
                 Time Limited Commands (TLC) feature set
                 Command Completion Time Limit (CCTL)
Security:
        Master password revision code = 65534
                supported
        not     enabled
        not     locked
        not     frozen
        not     expired: security count
        not     supported: enhanced erase
HW reset results:
        CBLID- above Vih
        Device num = 0 determined by CSEL
Checksum: correct
```

2. Modificación de los valores

Se puede modificar varios parámetros de los discos. Sin embargo, ¡cuidado! Algunas opciones de**hdparm** pueden resultar peligrosas tanto para los datos contenidos en el disco como para el propio disco. La mayoría de los parámetros son de lectura y escritura. Si no se especifica ningún valor,**hdparm** muestra el estado del disco (o del bus) para este comando. A continuación le presentamos algunos ejemplos de opciones interesantes.

- **-c**: anchura del bus de transferencia EIDE en 16 o 32 bits. 0=16, 1=32, 3=32 compatible.
- **-d**: utilización del DMA. 0=no DMA, 1=DMA activado.
- **-X**: modifica el modo DMA (mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5). Puede utilizar cualquiera de los modos anteriores o valores numéricos: 32+n para los modos mdma (n varía de 0 a 2) y 64+n para los modos udma.
- **-C**: modo de ahorro de energía en el disco (unknown, active/idle, standby, sleeping). Se puede modificar el estado con -S, -y, -Y y -Z.
- **-g**: muestra la geometría del disco.
- **-M**: indica o modifica el estado del Automatic Acoustic Management (AAM). 0=off, 128=quiet y 254=fast. No todos los discos lo soportan.
- **-r**: pasa el disco en sólo lectura.
- **-T**: bench de lectura de la caché del disco, ideal para probar la eficacia de transferencia entre Linux y la caché del disco. Hay que volver a ejecutar el comando dos o tres veces.
- **-t**: bench de lectura del disco, fuera de la caché. Mismas observaciones que para la opción anterior.

Así, el comando siguiente pasa el bus de transferencia a 32 bits, activa el modo DMA en modo Ultra DMA 5 para el disco sda:

```
# hdparm -c1 -d3 -X udma5 /dev/sda
```

Le mostramos a continuación otros ejemplos:

```
# hdparm -c /dev/sda

/dev/sda:
 IO_support    =  0 (default 16-bit)

# hdparm -C /dev/sda
```

```
/dev/sda:
 drive state is:  active/idle

# hdparm -g /dev/sda

/dev/sda:
 geometry      = 9729/255/63, sectors = 156301488, start = 0
p64p17bicb3:/etc/cups #

# hdparm -T /dev/sda

/dev/sda:
 Timing cached reads:   1320 MB in  2.00 seconds = 660.30 MB/sec

# hdparm -t /dev/sda

/dev/sda:
 Timing buffered disk reads:  168 MB in  3.03 seconds =  55.49 MB/sec
```

Elegir un sistema de archivos

1. Fundamentos

a. Definición de sistema de archivos

La acción de "formatear" un disco, un pendrive o cualquier soporte de datos consiste únicamente en crear en un soporte de memoria secundaria la organización lógica que permite colocar datos en él. La palabra "formateo" en Linux se utiliza para describir la creación de un sistema de archivos. Hablamos de un sistema de archivos que representa a la vez la organización lógica de los soportes tanto a un nivel inferior como a un nivel de usuario.

No se escribe la información en los discos de cualquier manera. Se requiere una mínima organización para colocar en ellos tanto la información relativa a los archivos como los datos almacenados. El sistema de archivos (y los controladores asociados) es el que define esta organización. Si bien los fundamentos organizativos suelen ser los mismos en los diferentes sistemas de archivos presentes soportados por Linux, las implementaciones y organizaciones lógicas de los datos en el disco varían bastante de uno a otro. De esta manera, no hay un único tipo de sistema de archivos, sino varios, puestos a disposición del usuario, el administrador o el ingeniero.

El principio básico es asociar un nombre de archivo con su contenido y autorizar su acceso: creación, modificación, supresión, desplazamiento, apertura, lectura, escritura, cierre. Conforme a este principio, el sistema de archivos debe gestionar lo que deriva de ello: mecanismos de protección de los accesos (permisos, propietarios), accesos concurrentes, etc.

b. Representación

Además de la organización y el almacenamiento de la información y datos en los archivos, el sistema de archivos debe facilitar al usuario una visión estructurada de sus datos, que permite distinguirlos, encontrarlos, tratarlos y trabajar con ellos, por ejemplo en forma de archivos dentro de una estructura de directorios con los comandos asociados. Asimismo, cada sistema de archivos debe proporcionar lo necesario para que los programas puedan acceder a él.

Un sistema de archivos Unix se organiza en forma de un árbol de directorios y subdirectorios desde una raíz común. Es una estructura en árbol. Cada directorio forma parte de una organización y propone, a su vez, una organización: el sistema de archivos dispone de una jerarquía ordenada. Se puede repartir la propia estructura en árbol entre varios soportes y sistemas de archivos.

c. Los metadatos

Las propiedades de un archivo se denominan metadatos, aunque en Linux recibe el nombre de inodo. Los inodos o metadatos de todo archivo están ubicados en un lugar especial del soporte de almacenamiento. El contenido de los metadatos cambia de un sistema de archivos a otro. Sin embargo, en Linux, encontramos siempre este contenido:

- los permisos;
- las últimas fechas de acceso y modificación;
- el propietario y el grupo;
- el tamaño;
- el número de bloques utilizados;
- el tipo de archivos;

- el contador de vínculos;
- un árbol de direcciones de bloques de datos.

d. Los nombres de los archivos

Los nombres pueden tener una longitud de 255 caracteres. La extensión no es relevante como componente del sistema de archivos a diferencia de lo que ocurre en Windows. La extensión, que más bien podemos llamar el sufijo, permite solamente diferenciar los nombres de los archivos en función del contenido supuesto. Se determina el tipo de archivo según su contenido, en particular el de los primeros bytes, que permiten determinar el tipo MIME. El comando **file** procede de esta manera. La extensión es una simple terminación del nombre del archivo, incluida en los 255 caracteres.

Los nombres de los archivos Unix no se establecen en los metadatos, sino en una tabla de catálogo. Gracias a ello es posible dar varios nombres a un mismo archivo.

e. El archivo de registro

Los actuales sistemas de archivos disponen a menudo de mecanismos que permiten garantizar en la medida de lo posible la integridad de los datos. El sistema más habitual es el "journaling" (es un anglicismo). El sistema de archivos mantiene al día un archivo, el registro, en general de un tamaño dado y circular (la nueva información termina por chafar la antigua) en el cual registra todos los cambios antes de efectuarlos realmente. En caso de parada abrupta, el sistema repasa las grabaciones del registro y comprueba si se realizaron las operaciones. Si es necesario las vuelve a ejecutar. El registro contiene operaciones atómicas (n operaciones indivisibles) y, por lo tanto, incluso si éste no está completo, se asegura la coherencia de los datos o bien completando el registro cuando es posible, o bien dando un paso atrás. Por lo tanto, la reparación es mucho más fiable y rápida que en un sistema de archivos tradicional.

2. Los sistemas de archivos en Linux

a. ext2

Se considera que el *second extended filesystem* ext2 es el sistema de archivos histórico de Linux, aunque éste utilizaba el MinixFS inicialmente. La primera versión, llamada ext (extended filesystem), aunque corregía los defectos de minix, tenía unos límites que impedían que fuera un verdadero sistema de archivos Unix. Ext2 es, por lo tanto, el primer sistema de archivos desarrollado específicamente para Linux, de un nivel de producción y conforme a las normas Unix (se habla de nivel de producción para indicar que un sistema cualquiera responde a los criterios de puesta en producción -uso real- en una empresa). Previsto desde el principio para soportar el aumento de funcionalidades, se sigue utilizando y mejorando desde 1993. Ext2 no es transaccional.

Aunque disponga de un sucesor (ext3), se sigue utilizando e incluso aconsejando en algunos casos. Es rápido y necesita menos escritura que los demás y, por lo tanto, conlleva menos desgaste de los soportes de almacenamiento, en particular los discos SSD, los pendrives o las tarjetas memoria. A veces estos soportes sólo pueden soportar un número restringido de ciclos de lectura/escritura...

Los archivos pueden tener un tamaño de hasta 2 TB (2048 GB), mientras que una partición puede alcanzar 32 TB, incluso 128 TB, según el tamaño de los bloques y la arquitectura.

b. ext3

El *third extended filesystem ext3* es el sucesor de ext2 desde 1999. Cuenta con un registro. Sobre todo, es totalmente compatible con ext2. El archivo de registro es una extensión de ext2. Es

posible utilizar un sistema de archivos ext3 como ext2, con los mismos comandos, las mismas operaciones. Es posible transformar en unos segundos un sistema ext2 en ext3, y viceversa. Este fue el sistema de archivos más utilizado durante cerca de 10 años. Y se encuentra todavía en muchas versiones de Linux en empresas.

Como para ext2, el tamaño máximo de los archivos es de 2 TB, y el de una partición, de 32 TB, siguiendo las mismas restricciones.

c. ext4

El conocido como *fourth extended filesystem ext4* es el sucesor de ext3. Estable desde 2008, es el sistema de archivos predeterminado de un gran número de distribuciones desde 2010. Durante su fase de puesta a punto se le conocía como ext4dev. Es compatible con ext3 y, aunque es avanzado, es considerado como el sistema de archivos de transición ante la llegada de **btrfs**. Como ext3, tiene diario. Añade nuevas funcionalidades, especialmente el concepto de extents, una técnica de preasignación de bloques en zonas contiguas que permite limitar la fragmentación de archivos y del disco y, por tanto, mejora el rendimiento. La rapidez de las comprobaciones del sistema de archivos ha sido revisada y ahora es incluso más rápido que ext3.

Un sistema de archivos ext3 puede montarse como ext4. Sin embargo, una partición ext4 sólo puede montarse como ext3 (o ext2) si el sistema de archivos ha sido creado sin el uso de extents.

El tamaño máximo de los archivos es de 16 TB y el de una partición de 1024 PB (Petabytes), es decir 1 EB (1 Exabyte).

d. btrfs

El sistema de archivos btrfs, que se pronuncia "butterFS" o "BetterFS", es un sistema de archivos que todavía se considera experimental (mientras se escribe este libro). Terminará por ser el nuevo sistema de archivos predeterminado de Linux. Inicialmente desarrollado por Oracle, btrfs está en desarrollo continuo. Además de tener como objetivo remplazar a ext4, se presenta como un competidor directo de xfs. Algunos lo usan ya habitualmente incluyendo sistemas empleados en producción. Utiliza el concepto de extent como ext4 y el de subvolúmenes: la creación de árboles separados en el interior de un sistema de archivos, con opciones propias en cada árbol (permisos, cuotas, etc.). Este método también permite la creación de instantáneas (*snapshots*) de un árbol, la imagen exacta del árbol en un instante determinado, permitiendo comprobar las modificaciones o volver al estado inicial, como con LVM. btrfs establece un sistema de protección de datos y de metadatos, especialmente con checksums que permiten averiguar si un archivo se ha corrompido.

Su flexibilidad le permite añadir y eliminar volúmenes y por ende redimensionar el sistema de archivos en caliente, soportando métodos RAID (0,1,5). Este es un punto fuerte en comparación con los sistemas de archivos ext.

Un sistema de archivos btrfs se puede organizar en varios volúmenes (discos o particiones), lo que le hace parecerse a soluciones del tipo LVM.

btrfs representa el estado del arte como sistema de archivos para equipos de cliente y servidores.

e. reiserfs

reiserfs fue el primer sistema de archivos integrado en Linux, antes incluso de ext3. Su fuerza reside, (además de en su archivo de diario) en la organización indexada de las entradas de los directorios (las tablas de catálogos que contienen las asociaciones inodos/archivos) y el manejo de los archivos de pequeño tamaño. Sus resultados son excepcionales cuando hay miles de archivos, de poco a mediano volumen. Puede ser redimensionado en un momento. El problema más relevante es que, con archivos muy grandes, es lento.

Los archivos pueden alcanzar 8 TB, y las particiones, 16 TB. Los nombres de archivos pueden tener 4032 caracteres, pero Linux los limita a 255 caracteres (concretamente con el **VFS**, *Virtual Filesystem Switch*).

A pesar de las evidentes mejoras que presenta, reiserfs es mucho menos utilizado. La razón estriba en que no es posible convertir un sistema de archivos ext2/ext3 en reiserfs ni viceversa, y eso le perjudica, ya que ext2/ext3 es omnipresente. reiserfs y su sucesor reiserfs4 han caído un poco en desuso debido a los problemas judiciales de su creador, en la cárcel durante varios años, sin ofrecer, por tanto, el mantenimiento de su sistema de archivos. reiser4 no se incorpora a la rama oficial del núcleo de Linux hasta la versión 3.12. El desarrollo de estos sistemas de archivos continúa gracias a los antiguos empleados de Namesys, patrocinados por DARPA.

f. xfs

xfs es el más antiguo de los sistemas de archivos transaccional en Linux, data de 1993. Creado por Silicon Graphics (sgi), fue llevado a Linux en 2000 y es hoy en día propiedad de Red Hat. Además de sus capacidades de almacenamiento aún impensables a día de hoy, tiene un sistema de registro transaccional muy eficaz y mecanismos avanzados, como la desfragmentación en línea (en caliente y al mismo tiempo que la escritura), la capacidad de efectuar snapshots (fijar el estado de un sistema de archivos en un instante t para restaurarlo más tarde), el dimensionamiento en caliente, la reserva de ancho de banda habilitada en comunicaciones para las entradas y salidas, etc.

El tamaño máximo (teórico, porque aún hoy nadie ha creado ninguno así de grande) de los archivos es de 8 EB (Exabytes). 1 EB equivale a 1024 PB (Petabyte), por lo tanto 1048576 TB, o sea, convertido a una unidad más comprensible, unos 100.000 millones de DVD. La partición puede alcanzar 16 EB, o sea, la capacidad máxima de un controlador en 64 bits. En sistemas de 32 bits, los tamaños se "limitan" a 16 TB.

El uso de xfs no está aún muy extendido en Linux, quizá por culpa de su supuesta complejidad para una configuración avanzada. Sin embargo, Red Hat incluye el soporte de xfs a partir de la versión 5.4 de su distribución (sin las herramientas, no obstante), y un soporte completo en la versión 6, como añadido, y es de hecho su sistema de archivos predeterminado en su versión 7. xfs y btrfs son pues competencia directa.

g. vfat

vfat (*Virtual File Allocation Table*) es un término genérico que agrupa las diferentes versiones de FAT que soportan los nombres de archivos largos (255 caracteres) en Windows. Se conservan estos sistemas de archivos y se siguen utilizando por razones a la vez históricas y prácticas. La mayoría de los soportes removibles, discos externos, pendrives y lectores MP3 utilizan un sistema de archivos de este tipo, debido a que:

- Se trata de un sistema de archivos adaptado a los pequeños volúmenes.
- Es un sistema de archivos simple de implementar, ideal para lectores multimedia.
- Compatibilidad entre diversas plataformas (Windows, Linux, BSD, Mac OS, etc.).

vfat padece defectos inherentes a su concepción:

- Se almacena el conjunto de la información dentro de una tabla única, incluyendo el nombre del archivo y cada dirección y longitud de los bloques (llamados clústeres) que componen los datos del archivo.
- De hecho, FAT intenta agrupar los datos de un archivo en el mayor número de clústers contiguos del soporte. En caso de que se hagan muchas modificaciones (añadido, supresión, etc.), el sistema se ve fuertemente fragmentado.
- Por otro lado, cuanto más grande es el soporte, más lento va FAT, ya que debe comprobar toda la tabla FAT para encontrar clústeres disponibles.
- La gestión de nombres largos es puro bricolaje para muchas personas, ya que FAT continúa asegurando la compatibilidad (aún hoy en día) con los nombres cortos en 8.3.
- A diferencia de los demás sistemas de archivos de Unix, Linux o Windows recientes, FAT no gestiona ningún atributo extendido, en particular ninguna noción de los permisos y

propietarios.

- FAT está limitado a un tamaño de archivo de 4 GB. Se deben cortar los archivos más grandes (base de datos, archivos, imágenes, ISO de DVD, etc.) en consecuencia y los programas (captura de audio / vídeo, sgbd, etc.) deben gestionar esta limitación.

Aunque VFAT sea todavía el sistema de archivos predeterminado de muchos lectores multimedia, smartphones, pendrives, tarjetas de memoria y dispositivos de fotografía digital, cada vez pierde más protagonismo en relación a NTFS en el caso de los soportes de intercambio y ext3/ext4 (por ejemplo en Android). La llegada de la alta definición amenaza con la desaparición de VFAT debido a que su tamaño de archivo máximo es de 4 GB.

El mayor problema de VFAT no es técnico, sino jurídico. Aunque sea simple, antiguo y esté extendido, Microsoft es el propietario gracias a una patente registrada en 1994 (año de lanzamiento de Windows 95 y, por tanto, VFAT) en la Oficina Europea de Patentes y aceptada en diciembre de 2001 (y 2006 en los Estados Unidos). Microsoft ha llevado ante los tribunales a muchas empresas por violación de su patente, para cobrar los royalties. Estos procedimientos han provocado, por ejemplo, que algunos fabricantes utilicen MTP en Android, o que Tomtom deje de utilizar VFAT para sus GPS. Muchos procedimientos en curso en Europa tratan de invalidar está patente, lo que llevará probablemente a que el software no pueda patentarse.

Linux gestiona perfectamente VFAT. Pero su uso en soportes compartidos entre Windows y Linux tiene menos sentido: el que exista un soporte completo de NTFS en Linux permite utilizar estos sistemas de archivos, que tiene mejor rendimiento.

Particionado

1. Particionado lógico

En este manual consideraremos soporte de almacenamiento a cualquiera de tipo magnético o de memoria como, por ejemplo, discos duros, SSD, tarjeta de memoria, etc. Es decir, todo lo que se pueda parecer a un disco duro según la visión clásica: un espacio de datos que se puede seccionar en varias entidades lógicas e independientes y que dispone cada una de su propio sistema de archivos.

Se puede considerar un disco como una larga banda de espacio de almacenamiento dividida en casillas que pueden contener una cantidad determinada de información. Se puede utilizar el disco tal cual, como espacio de almacenamiento. Nada impide crear un sistema de archivos en un disco sin pasar por la etapa de particionado. Sin embargo, es importante dar una organización lógica a este espacio y a los sistemas de archivos que va a contener, aunque se trate sólo de separar los datos (los archivos de datos) de los tratamientos (los programas que los utilizan y el sistema).

El particionado consiste en una división lógica del disco. Se fracciona el disco físico, real, en varios discos virtuales, lógicos: las particiones. Se ve cada partición como un disco independiente que contiene su propio sistema de archivos.

Existen dos métodos de particionado: MBR, previsto para los equipos basados en BIOS, y GPT, para los equipos basados en UEFI. Estas últimas son también compatibles con el particionado BIOS, pero no al revés.

2. Particionado MBR

a. MBR y BIOS

El método de particionado MBR se llama de esta forma porque la partición principal se ubica en este sector específico del disco. El particionado MBR data de los años 1980, cuando los PC trabajaban en 16 bits, y se le han hecho cuatro apaños, para trabajar en 32 bits. La posición de un bloque se codifica en 32 bits, y el tamaño de un bloque es de 512 bytes. Podemos entonces calcular el tamaño máximo de un disco: 2^{32}*512 bytes: 2 TB. No es posible emplear este método con discos de más de 2 TB.

Este método de particionado sigue siendo el más empleado, pero la disponibilidad de equipos basados en UEFI, los nuevos sistemas operativos y los discos de capacidades superiores a 2 TB lo dejarán obsoleto en unos pocos años.

b. MBR

El primer sector es el **MBR**, *Master Boot Record* o zona de arranque. Con un tamaño de 512 bytes, contiene en sus primeros 446 bytes una rutina (un programa) de arranque destinado a ejecutar el sistema operativo desde la partición activa, o desde el gestor de arranque (bootloader). Los 6 últimos bytes de este bloque, opcionales, son usados para firmar el disco con 4 bytes. A continuación, 2 bytes nulos, y más adelante, los 64 bytes que contienen la tabla de las cuatro particiones primarias. El conjunto termina con una firma 0xAA55 en dos bytes.

Formato de un MBR

c. Las particiones

Una partición es una división lógica del disco. Existen tres tipos:

- Las particiones primarias, en un total de cuatro, son las descritas en el MBR.
- Las particiones extendidas (primarias extendidas), una sola por disco (aunque en teoría sea posible crear particiones extendidas dentro de otras particiones extendidas).
- Las particiones lógicas.

Un disco puede contener hasta 63 particiones en IDE, 15 en SCSI (es un límite de la implementación oficial del SCSI) o mediante la librería libata. El límite actual es de 15 particiones para todos los discos con los últimos núcleos y la API libata. Sin embargo, algunas distribuciones permiten utilizar la antigua API (PATA) para volver al antiguo sistema.

Observe que se trata efectivamente de un límite por disco, y no del número total de particiones gestionadas por el sistema.

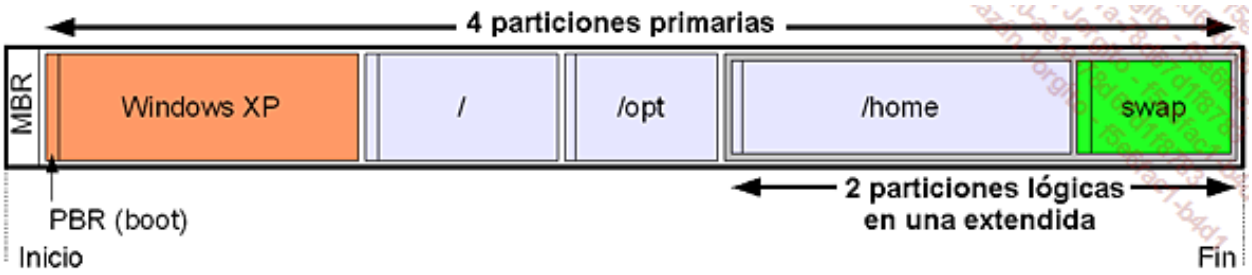
Para superar el límite de las 15 particiones, es posible utilizar el "device mapper" de Linux, que requiere en particular la gestión del **LVM** (*Logical Volume Management*). El LVM permite agrupar varios discos (Physical Volumes) en una sola unidad (Volume Group) vista por el sistema como un enorme disco único que puede dividir en particiones (Logical Volumes) sin limitación de número. Además, puede añadir discos en el grupo después, aumentar o reducir el tamaño de las particiones sin preocuparse de su ubicación física real...

Se numeran las particiones de 1 a n (15 o 63). Una partición con un valor superior o igual a 5 indica que se trata obligatoriamente de una partición lógica. Como sólo puede haber cuatro particiones primarias, se suele crear la última (la 4) como extendida:

- Particiones 1 a 3: primarias
- Partición 4: extendida
- Particiones 5 a n: lógicas

El número de la partición aparece a continuación del nombre del archivo periférico de disco:

- hda1: primera partición primaria del primer disco IDE;
- hdb5: quinta partición, primera partición lógica del segundo disco IDE;
- sda3: tercera partición primaria del primer disco SCSI / libata;
- sdc8: octava partición, o sea, cuarta partición lógica del tercer disco SCSI/libata.



Descripción esquemática de un disco

d. EBR

Como cada partición extendida debe describir las particiones lógicas que contiene, debe disponer también de una tabla de partición. El **EBR** (*Extended Boot Record*) retoma la estructura del MBR, excepto porque sólo hay dos grabaciones posibles en la tabla de las particiones. La primera indica efectivamente la posición y el tamaño de una partición lógica, mientras que la segunda está vacía si

es la única partición lógica, o apunta a otro EBR. Por lo tanto, puede haber varios EBR en una partición extendida.

- Los EBR forman una lista encadenada, la segunda entrada de partición apuntando al EBR siguiente.
- Sólo hay una partición lógica descrita por EBR.

e. PBR

El **PBR** (*Partition Boot Record*), también llamado **VBR** (*Volume Boot Record*) o Partition Boot Sector es el primer sector de cada partición primaria o lógica. Puede contener una rutina de inicio de un sistema operativo, un cargador de inicio o incluso nada si la partición no tiene como misión iniciar un sistema operativo. Cuando el MBR no contiene rutinas, la BIOS intenta iniciar y ejecutar la rutina del PBR de la partición marcada como activa.

f. Tipos de particiones

Cada partición dispone de un tipo que permite determinar su contenido. Es un identificador numérico codificado en un byte generalmente presentado en hexadecimal. Parece importante facilitar una lista aquí para que pueda entender correctamente la finalidad de este identificador.

```
 1  FAT12           24  NEC DOS         81  Minix / old Lin bf  Solaris
 2  XENIX root      39  Plan 9          82  Linux swap / So c1  DRDOS/ sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c4  DRDOS/ sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/ sec (FAT-
 5  Extended        41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 6 FAT16            42  SFS             86  NTFS volume set da  Non-FS data
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .
 8  AIX             4e  QNX4.x 2nd part 88  Linux plein tex de  Dell Utility
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e1  DOS access
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          e4  Speed-Stor
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 f  W95 Etendu (LBA 54  OnTrackDM6      a5  FreeBSD         ee  EFI GPT
10  OPUS            55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f0  Linux/ PA-RISC b
12  Compaq diagnost 5c  Priam Edisk     a8  UFS Darwin      f1  Speed-Stor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f4  Speed-Stor
16  Hidden FAT16    63  GNU HURD or Sys ab  Amorce Darwin   f2  DOS secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fd  Linux raid auto
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fe  LANstep
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid ff  BBT
1c  Hidden W95 FAT3 75  PC/IX
```

Como el tipo de partición debería reflejar el sistema de archivos que contiene, una partición de tipo 0x0c debería contener un sistema de archivos de tipo FAT32 LBA (discos grandes). Una partición de tipo 0x83 debería contener un sistema de archivos Linux. Pero ¿cuál? Ya ha visto que existen varios...

Observe el predominio de los tipos de partición para Windows. Windows se basa esencialmente en este tipo para determinar su contenido. Nada impide crear una partición de tipo Linux y colocar en ella un sistema de archivos FAT32. Sin embargo, si lo hace, Windows no reconocerá la partición (considerada de tipo desconocido) y no podrá acceder al contenido.

Linux reconocerá en general (hay una pocas excepciones) el contenido de una partición por el sistema de archivos que reside en ella. Puede crear un sistema de archivos ext3 en una partición de tipo 0x0e y constatar que todo funciona. El tipo 0x83 puede acoger cualquier sistema de archivos Linux: ext2, ext3, reiserfs, jfs, xfs... Sin embargo, por razones de compatibilidad, tenga cuidado en respetar la asociación tipo de partición <-> sistema de archivos y ser coherente.

3. Particionado GPT

a. GPT y UEFI

GPT significa *GUID Partition Table*, **GUID** significa *Globally Unique Identifier*. Este es un estándar que permite describir la tabla de particiones de un disco duro, que forma parte de las especificaciones de **UEFI***Unified Extensible Firmware Interface*. Para resumir:

- UEFI substituye a la BIOS.
- GPT reemplaza al particionado MBR.

Ha constatado que el diseño de una BIOS y su método de direccionamiento de un disco duro, basado en valores codificados en 32 bits, impiden el uso de discos de más de 2 TB y reducen el número de particiones posibles. La disponibilidad de soportes de almacenamiento de mayores capacidades, el nuevo hardware y los nuevos sistemas operativos han adelantado la necesidad de disponer de un sistema de arranque y de particionado más moderno.

El desarrollo inicial de GPT se remonta a finales de los 90 y lo inició Intel, dentro de EFI. Su evolución a UEFI y su aceptación como estándar para los constructores y fabricantes en 2014 lo convirtió en 2010 en el habitual para todo el hardware (placas base) y sistemas operativos recientes. Todas las placas base basadas en UEFI aceptan GPT.

Por razones de compatibilidad, es posible en ciertos casos emplear GPT con algunas BIOS, especialmente para el empleo de una partición de arranque particular, la **BIOS Boot partition**, vinculada a un gestor de arranque particular, como GRUB2.

El tamaño de dirección de un bloque lógico está codificado en 64 bits. El tamaño de un bloque lógico es al menos de 512 bytes pero puede ser más grande. El tamaño máximo de un disco con un tamaño de bloque de 512 bytes es de 2^{64}-1*512 bytes: 8 ZiB (Zebibyte), o sea 8 mil millones de terabytes.

No contamos más la posición en sectores o bloques, pero en términos de **LBA** (*Logical Block Addressing*, direccionamiento por bloques lógicos). Empleamos entonces LBA 0 para los 512 primeros bytes, LBA 1 para los siguientes 512, y así sucesivamente.

Todas las versiones actuales de las distribuciones Linux soportan GPT.

b. GUID

GUID, identificador global único, es un valor codificado en 128 bits que sirve como identificador único para un componente software. No podemos saber si un GUID es realmente único (en el mundo en todo caso), pero la probabilidad, vistos los valores del conjunto, de que dos valores aleatorios de 128 bits sean iguales en componentes software cercanos o que se espera que trabajen juntos es casi nula.

GUID es equivalente al UUID, nombre con el que es más conocido en Linux y Unix. Observe que en su forma hexadecimal, un GUID se parece a esto: 41649fe8-9341-4bce-977a-687f9da63fdb

Distinguimos:

- Un bloque de cuatro bytes (32 bits).
- Tres bloques de dos bytes (3 veces 1 bits, 48 bits).
- Un bloque de seis bytes (48 bits).

En GPT, el GUID sirve para describir:

- El identificador único del disco.
- El identificador del tipo de partición (equivalente a los tipos de particiones MBR).
- El identificador único de una partición.

c. LBA 0

El **LBA 0**, o sea los primeros 512 bytes del disco, conservan su rol de MBR, pero son conocidos como MBR protector (*Protective MBR*). Su función, dependiente de su estructura, es la de impedir las escrituras en el disco por herramientas que no reconocen el formato GPT. Este MBR describe de hecho una partición, de tipo 0xEE, que abarca el conjunto del disco hasta un máximo de 2 TB, las antiguas herramientas y BIOS no son capaces de ver más allá. Este tipo de partición es desconocido para las herramientas del tipo MBR, no pueden modificar el disco, salvo borrando esta partición.

La presencia de un Protective MBR es la firma de un disco GPT.

Tratándose sin embargo de un MBR, un BIOS reciente puede ser programado para reconocer este tipo de "partición" y utilizar el cargador de arranque (bootloader). En este caso, en Linux con GRUB2, este fin de código debe finalmente cargar el contenido de otra partición especial, la BIOS Boot partition.

d. LBA 1

La cabecera GPT está en la posición LBA 1. Existe una copia en LBA -1, a partir del final del disco. Esta cabecera, firmada 45 46 49 20 50 41 52 54, que se traduce en "PARTE EFI" en ASCII, contiene información sobre la estructura GPT:

- Su propia ubicación y tamaño,
- La ubicación y el tamaño de la cabecera sub (al final del disco)
- Asociada a controlar el dinero,
- Las primeras y últimas direcciones utilizadas para las particiones,
- El GUID del disco,
- La dirección de la tabla de particiones (por lo general LBA 2)
- El número y el tamaño de las particiones descriptores,
- La suma de comprobación de la tabla de particiones.

El conjunto ocupa 92 bytes, los restantes 420 se rellenan con ceros.

La presencia de las sumas de comprobación permite que el firmware UEFI, el gestor de arranque o las herramientas especializadas compruebe el estado de la cabecera y su copia. En caso de error en el LBA 1, LBA -1 puede ser utilizado. En caso de error en ambos, el disco queda inutilizable. Es lo mismo para el control de las tablas de partición.

e. LBA 2 a 33

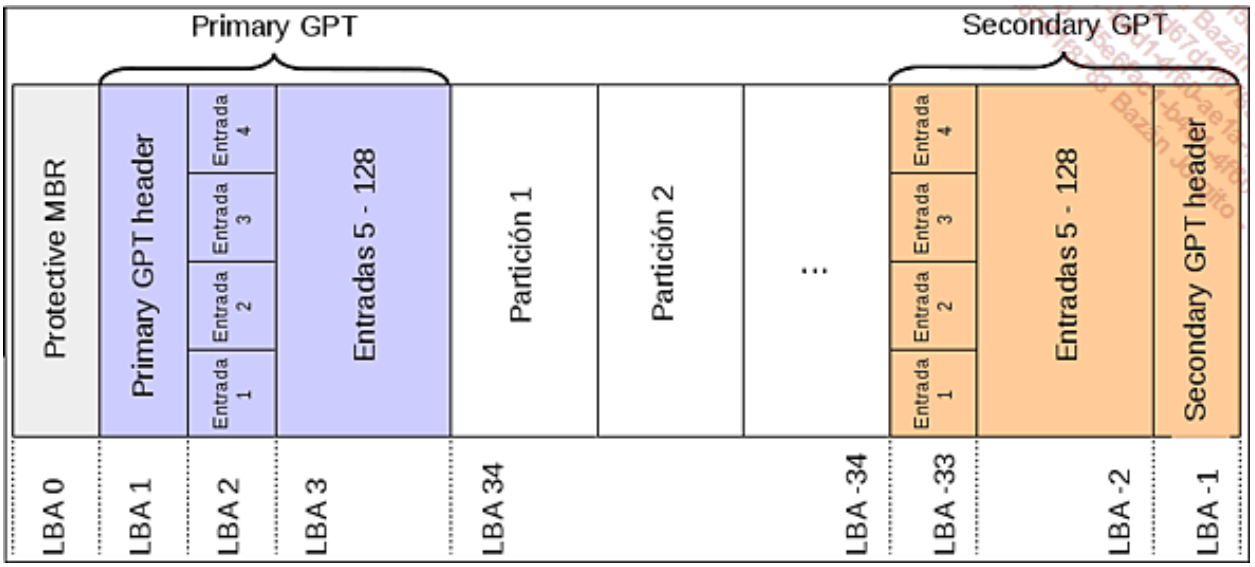
Los LBA 2 a 33 contienen los descriptores de las particiones. Cada partición está descrita en 128 bytes, donde 4 son para LBA. Los 32 LBA permiten describir 128 particiones. Este es el mínimo impuesto por el estándar UEFI, el número y el tamaño de los descriptores de las particiones pueden ser modificados en el LBA 1.

Una copia de seguridad de los descriptores se encuentra en LBA -2 a -33 (fin del disco).

El descriptor de partición está estructurado como sigue:

- el tipo GUID de la partición (16 bytes), el equivalente del tipo de partición MBR,
- el GUID de la partition misma (16 bytes),
- el LBA de inicio de la partición (8 bytes),
- el LBA del final de la partición (8 bytes),
- los atributos (por ejemplo, solo lectura) de la partition (8 bytes),
- el nombre de la partición en formato UTF-16 (2 bytes por caracter, 36 caracteres, 72 bytes).

Así llegamos a un tamaño de 128 bytes.



Estructura de un disco GPT

f. Tipos de particiones

Los tipos de particiones están descritos por los GUID. Al ser el tamaño de los GUID imponente, aquí solo listamos los GUID más frecuentes para las particiones previstas para Linux.

Type	GUID
Linux filesystem data	0FC63DAF-8483-4772-8E79-3D69D8477DE4
RAID partition	A19D880F-05FC-4D3B-A006-743F0F84911E
Swap partition	0657FD6D-A4AB-43C4-84E5-0933C84B4F4F
LVM partition	E6D6D379-F507-44C2-A23C-238F2A3DF928

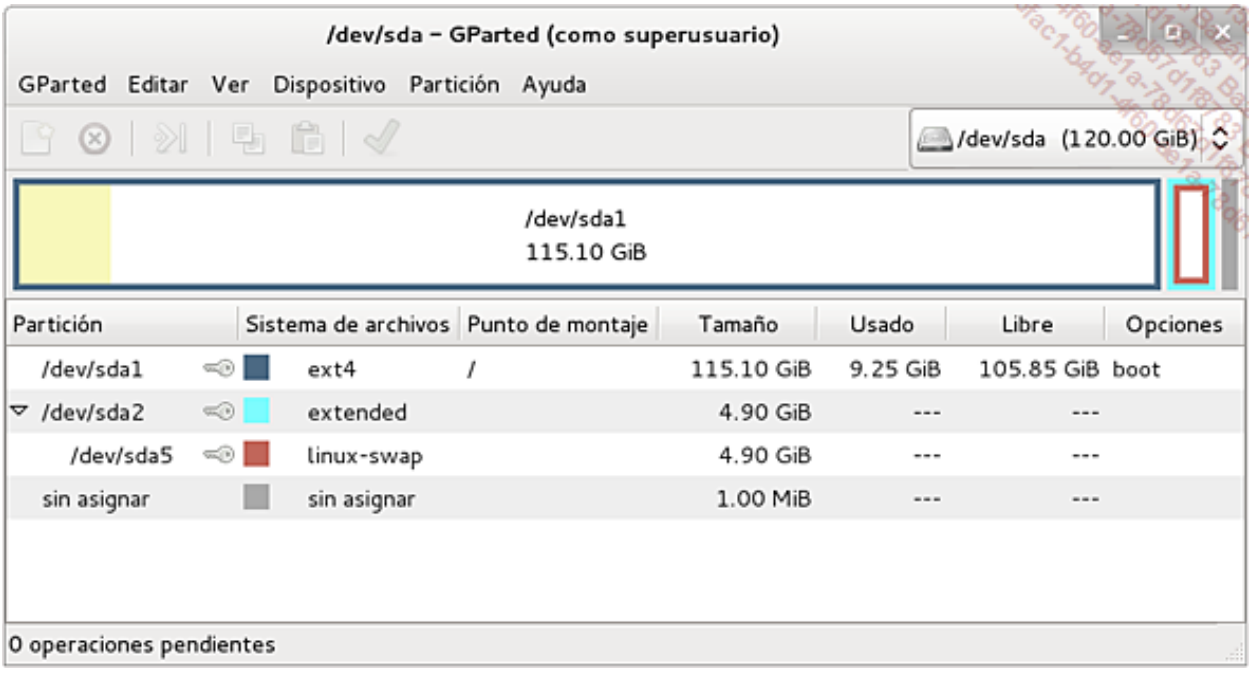
4. Manejar las particiones

a. Herramientas de gestión de particiones

Las herramientas **fdisk**, **cfdisk**, **sfdisk**, **parted** o también **gdisk**, sin contar con las herramientas gráficas disponibles durante la instalación o en los paneles de configuración, permiten manejar las particiones.

- **fdisk** es la más antigua y más utilizada de las herramientas de particionado. No tiene relación con el fdisk de Microsoft. Se basa en menús y atajos textuales.
- **cfdisk** es un poco más «visual» y se utiliza con las flechas direccionales. Permite las mismas operaciones que fdisk y es de fácil manejo.
- **sfdisk** funciona, opcionalmente, de forma interactiva. Es bastante complicada, pero más precisa.
- **parted** permite operaciones muy avanzadas en las particiones, como, por ejemplo, su redimensionamiento. Presenta una interfaz interactiva (intérprete de comandos) que atiende a scripts. Es compatible con GPT. Pero, además, hay en el mercado interfaces gráficas de parted, como qtparted o gparted.
- **Gdisk** es equivalente a fdisk para GPT.

En la captura siguiente puede ver a **gparted** en acción.



gparted, un editor de particiones gráfico

La siguiente sección describe las operaciones de particiones de tipo MBR. La sección Manipular las particiones GPT ofrece las diferencias entre GPT mediante el uso de gdisk.

b. Manipular las particiones MBR

Listar

Los administradores e ingenieros de sistemas suelen utilizar la herramienta **fdisk**, que es a la vez la más antigua y más estándar. Hay que tener privilegios de root para poder usar fdisk.

```
fdisk [-l] [disco]
```

Cada parámetro es opcional. Iniciado tal cual, **fdisk** se sitúa en el primer disco del sistema. El parámetro `-l` permite listar las particiones del disco dado o de todos los discos. La información obtenida es la misma que en modo interactivo con la entrada p (print) del menú.

```
# fdisk -l /dev/sda

Disco /dev/sda: 164,6 Gb, 164696555520 bytes
255 heads, 63 sectors/track, 20023 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000c02ae

Periférico Arranque  Principio    Fin      Bloques   Id  Sistema
/dev/sda1               1        5222    41945683+   7  HPFS/NTFS
/dev/sda2   *        5223        5288      530145   83  Linux
/dev/sda3            5289       10510    41945715   83  Linux
/dev/sda4           10511       20023    76413172+   f  W95 Extendido (LBA)
/dev/sda5           10511       10772     2104483+  82  Linux swap / Solaris
/dev/sda6           10773       20023    74308626   83  Linux
```

Los campos hablan por sí mismos. Fíjese en que la partición sda4 es la extendida. Tiene como tipo 0x0f, pero cualquier tipo extendido hubiese funcionado: se consideran los tipos 0x05 y 0x85 como idénticos. Sin embargo, es posible que Windows no reconozca estos tipos ni, por lo tanto, las particiones lógicas contenidas en ellos, en particular las de un disco LBA. Por eso, la mayoría de las herramientas de particionado de las distribuciones Linux prefieren el tipo asociado a Windows.

Sólo se tienen en cuenta las operaciones efectuadas con **fdisk** al final, una vez que haya guardado usted sus modificaciones, y no al mismo tiempo. Si piensa que se ha equivocado, no dude en salir sin guardar o en hacer un [Ctrl] **C**. Se perderán sus modificaciones, pero habrá salvado sus particiones.

Lista de particiones

El ejemplo siguiente se basa en un disco reconocido por el sistema como /dev/sdb que no contiene ninguna partición. El objetivo es crear tres particiones: una primaria, una extendida y una lógica.

- Ejecute **fdisk** con el disco como argumento. No tenga en cuenta las primeras líneas visualizadas, excepto si indican un error.

```
# fdisk /dev/sdb
...
Comando (m para la ayuda):
```

- Verifique primero la existencia de particiones con la tecla p (print), luego [Entrar].

```
Comando (m para la ayuda): p

Disco /dev/sdb: 4026 Mb, 4026531840 bytes
64 heads, 62 sectors/track, 1981 cylinders
Units = cilindros of 3968 * 512 = 2031616 bytes
Disk identifier: 0x0003ed63

Periférico  Arranque   Principio   Fin     Bloques   Id  Sistema
/dev/sdb1   *          1           1981    3930273   c   W95 FAT32 (LBA)
```

Suprimir

Para suprimir una partición, utilice la tecla d (delete); luego, si hay varias particiones, el número de partición (sdbX, siendo X el número). Si hay una única partición, se coge por defecto.

```
Comando (m para la ayuda): d
Partición seleccionada 1
```

Crear

Para crear una partición, utilice la tecla n (new). Luego debe elegir el tipo de partición: primaria o extendida.

```
Comando (m para la ayuda): n
Acción de comando
   e   extendida
   p   partición primaria (1-4)
```

- Para esta primera partición, seleccione una partición primaria con la tecla p (que significa "primary" esta vez).
- Como el MBR contiene cuatro entradas, puede elegir el número de partición que crear. Es perfectamente posible crear la partición sdb2 antes de la sdb1. Aquí, teclee **1**.
- El primer cilindro corresponde a la posición de principio de su partición. Por defecto, fdisk se coloca en el primer cilindro disponible desde el principio del disco. Es perfectamente posible crear particiones primarias en la mitad de un disco. Seleccione aquí el valor por defecto (1) pulsando [Entrar].
- Finalmente, elija el tamaño de la partición. Es preferible utilizar una unidad legible, como los KB

o MB. Por ejemplo, para una partición de 1 GB, es decir, de 1024 MB, use **+1024M** y pulse [Entrar]. Ahora se ha definido la partición.

```
Comando (m para la ayuda): n
Acción de comando
   e   extendida
   p   partición primaria (1-4)
p
Número de partición (1-4): 1
Primer cilindro (1-1981, por defecto 1):
Utilización del valor por defecto 1
Último cilindro o + tamaño o +tamañoM o +tamañoK (1-1981, por
defecto 1981): +1024M
```

- Compruebe el estado de la partición (p).

```
Comando (m para la ayuda): p
...
Periférico Arranque   Principio   Fin      Bloques     Id  Sistema
/dev/sdb1              1          505     1001889     83  Linux
```

Guardar

Salga de fdisk guardando su tabla de las particiones con la tecla w (write). Fdisk escribe la nueva tabla de las particiones en el MBR o los EBR. Puede que el sistema le muestre alertas que nosotros aquí le indicamos en negrita.

```
Comando (m para la ayuda): w
¡Se alteró la tabla de particiones!

Llamada de ioctl() para volver a leer la tabla de particiones.

ADVERTENCIA: la relectura de la tabla de particiones fracasó con
el error 16: Periférico o recurso ocupado.
El kernel va a seguir utilizando la antigua tabla.
Se utilizará la nueva tabla durante el próximo reinicio.
Sincronización de los discos.
```

Este mensaje significa que, como el disco está en uso, Linux no puede volcar la nueva tabla ni, por lo tanto, crear las nuevas particiones. El comando siguiente lo puede confirmar. Observe que la última línea debería mostrarle su nueva partición. Pero no es así.

```
# cat /proc/particiones | grep sdb
8    16    3932160 sdb
```

Forzar la sincronización

Para corregir este último problema y forzar al núcleo a leer de nuevo la tabla de las particiones, tiene a su disposición dos comandos. El primero es **blockdev** con el parámetro `--rereadpt` (re-read partition table).

```
# blockdev --rereadpt /dev/sdb
```

El segundo es **partprobe**, disponible solamente si parted está instalado. Puede probarlo si blockdev no funciona. Por defecto, vuelve a leer las tablas de todas las particiones, pero le puede especificar como argumento un disco en concreto.

```
# partprobe /dev/sdb
```

Compruebe si se reconoce la partición.

```
# cat /proc/particiones | grep sdb
8    16    3932160 sdb
8    17    1001889 sdb1
```

Ahora puede utilizar su nueva partición y añadirle un sistema de archivos. Cree ahora una partición extendida, luego una partición lógica. Observe que se crea una partición extendida o lógica de la misma manera que las otras. Una partición extendida no tiene por qué ser la última de las primarias. También puede ser la segunda, por ejemplo. Si no ha utilizado todo el tamaño para crearla, puede completar más adelante la creación de particiones primarias.

Se pueden usar los comandos de parted como **mkpart**, **rm** o **resize** para trabajar con las particiones. El comando **partx** también permite notificar en el núcleo las modificaciones realizadas a las tablas de partición.

Modificar el tipo

Como la modificación del tipo de una partición no implica la modificación de su tamaño, puede hacerlo en cualquier momento. Le presentamos a continuación el modo de proceder para pasar del tipo por defecto Linux al tipo FAT para que Windows reconozca la partición.

- Ejecute **fdisk** y pulse t (tipo).
- Seleccione la partición, la 5 para la primera partición lógica.

  Puede visualizar todos los tipos pulsando L, que le facilita la misma lista mencionada más arriba. Utilice el tipo c W95 FAT32 (LBA) para estar tranquilo.

- Guarde con w.

```
Comando (m para la ayuda): t
Número de partición (1-5): 5
Código Hex (teclear L para listar los códigos): c
Tipo de partición del sistema modificado de 5 a c (W95 FAT32 (LBA))

Comando (m para la ayuda): p

Disco /dev/sdb: 4026 Mb, 4026531840 bytes
64 heads, 62 sectors/track, 1981 cylinders
Units = cilindros of 3968 * 512 = 2031616 bytes
Disk identifier: 0x0003ed63

Periférico Arranque  Principio    Fin     Bloques    Id  Sistema
/dev/sdb1              1         505    1001889     83  Linux
/dev/sdb2            506        1981    2928384      5  Extended
/dev/sdb5            506        1981    2928353      c  W95 FAT32 (LBA)

Comando (m para la ayuda): w
```

Los tipos de particiones más utilizados para Linux son los siguientes:

- **83**: Partición de tipo Linux (datos)
- **82**: Partición de tipo swap
- **fd**: Partición de tipo RAID
- **8e**: Partición de tipo LVM

c. Manipular las particiones GPT

El comando **gdisk**, GPT fdisk es su verdadero nombre, es el equivalente de fdisk pero para un disco GPT. Su uso es cercano al de fdisk, y también su sintáxis. Veamos qué pasa con un disco virgen:

```
# gdisk /dev/sdb
GPT fdisk (gdisk) version 0.8.10
Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

 Creating new GPT entries.

 Command (? for help):
```

Ya no existen las nociones de particiones primarias, extendidas y lógicas. Las particiones se numeran de 1 a 128. El proceso de creación es idéntico al particionado con fdisk. La única diferencia notable es la lista de códigos y los tipos de particiones. Parece evidente que no va a introducir el GUID, también los ctypes ofrecidos están codificados con cuatro caracteres hexadecimales, y para los más utilizados, son los mismos que en fdisk, completados por dos ceros:

- **8300** : partición de tipo Linux (datos)
- **8200** : partición de tipo swap
- **fd00** : partición de tipo RAID
- **8e00** : partición de tipo LVM

Puede constatar la presencia de los códigos 8301 para una partición prevista para mantener un /home y los códigos ef02 para un tipo de partición boot BIOS, que volveremos a ver cuando abordemos la gestión del arranque.

No se efectúa ninguna modificación hasta que no escriba [w]. La tecla [c] permite cambiar el nombre de la partición.

Puede obtener información detallada de una partición con [i]:

```
Command (? for help): i
Using 1
Partition GUID code: 0FC63DAF-8483-4772-8E79-3D69D8477DE4 (Linux filesystem)
Partition unique GUID: 81CDF84E-D87B-48C5-A0C8-13EE07134D56
First sector: 2048 (at 1024.0 KiB)
Last sector: 10487807 (at 5.0 GiB)
Partition size: 10485760 sectors (5.0 GiB)
Attribute flags: 0000000000000000
Partition name: 'root'
```

Observe: el uso de gdisk en un disco particionado en el método MBR puede ser peligroso: al guardar el disco completo éste se convierte al format GPT, Protective MBR incluido.

Manejar los sistemas de archivos

1. Definición básica

a. Bloque

El bloque es la unidad básica, atómica, de almacenamiento del sistema de archivos. Un archivo ocupa siempre un número entero de bloques. Así, si el tamaño de un archivo es de un byte y el bloque en el que está tiene un tamaño de 4096 bytes, se desperdician 4095 bytes. De esta manera, es posible llenar un sistema de archivos con n archivos de 1 bytes, donde n representa el número de bloques, mientras que ¡el volumen total de los datos sólo es de n bytes!

Supongamos un disco que contiene 102.400 bloques de 4096 bytes. Su tamaño total es de 400 MB. Supongamos 102.400 archivos de 384 bytes. El tamaño total de los datos es de 37,5 MB. Ahora bien, el disco está lleno, ¡ya que todos los bloques están siendo utilizados! Hay 362,5 MB perdidos. Por lo tanto, es muy importante tener cuidado con el tamaño de bloques, sobre todo si los archivos que se van a almacenar son de pequeño tamaño.

Algunos comandos calculan el tamaño de los archivos en bloques, como du, df o find. Históricamente, el tamaño de un bloque era de 512 o 1024 bytes. Es una unidad cuyo valor puede cambiar según el comando, y a menudo sin relación con el tamaño de los bloques del sistema de archivos en el cual se trabaja. Se impone la prudencia.

b. Superbloque

Cada sistema de archivos dispone de al menos un superbloque. Un superbloque es una zona de metadatos que contiene varios datos sobre el sistema de archivos:

- su tipo;
- su tamaño;
- su estado;
- información (posición) en las demás zonas de metadatos (otros superbloques, tabla de inodos, etc.).

Linux intenta primero leer el superbloque primario, el primero del disco. Puede ocurrir que este último esté corrompido después de operaciones erróneas o una avería. En este caso, ya no se puede acceder a los datos del disco (es imposible saber, por ejemplo, dónde están los inodos). Un sistema de archivos de Linux dispone de copias (backups) de los superbloques en varios sitios del disco. Como las escrituras en los diversos superbloques son síncronas, son todos idénticos. Como último recurso, si se suprime uno de ellos, se puede volver a copiar desde otro.

A continuación verá cómo disponer de toda la información en un sistema de archivos ext2 o ext3.

c. Tabla de inodos

La palabra **inodo** es la contracción de "índice nodo", o sea nodo de índice. Es una estructura de datos que contiene la información que describe y representa un archivo. Son los **atributos** de un archivo. Cada archivo dispone de un número de inodo (i-number). Todos los inodos están presentes dentro de una tabla de inodos. Se suele repartir partes de esta tabla después de cada superbloque. Una tabla de inodo forma parte de los metadatos.

Un archivo sólo puede tener un único inodo. Un inodo es único dentro de un único sistema de archivos. Cada sistema de archivos dispone de una tabla de inodos independiente. Si el archivo titi lleva el número de inodo 12345 en un primer sistema de archivos, aunque pepito lleve el número de

inodo 12345 en otro, estos archivos no tiene relación entre sí.

Sin embargo, veremos más adelante que se puede asignar el mismo número de inodo a dos nombres de archivos dentro de un mismo sistema de archivos. Estos dos nombres representan entonces un solo y mismo archivo.

Contenido

El contenido de un inodo cambia de un sistema de archivos a otro, pero la norma POSIX impone que cada uno de ellos disponga de, al menos, los atributos siguientes para cada archivo:

- tamaño;
- identificador del periférico que lo contiene;
- propietario;
- grupo;
- número de inodo;
- modo (permisos) de acceso;
- fecha de última modificación de inodo (change time);
- fecha de última modificación de contenido (modification time);
- fecha de último acceso (access time);
- contador de hard links (vínculos físicos o duros, ver más adelante).

Un inodo no contiene el nombre del archivo.

Puede obtener información relativa a un inodo con el comando **stat**:

```
# stat capitulo4.doc
  File: `capitulo4.doc'
  Size: 199168          Blocks: 408        IO Block: 4096   archivo regular
Device: 811h/2065d      Inodo: 16629765    Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/     seb)   Gid: (  100/   users)
Access: 2008-03-17 21:39:31.000000000 +0100
Modify: 2008-03-17 21:39:30.000000000 +0100
Change: 2008-03-17 21:40:04.000000000 +0100
```

Direcciones

El inodo contiene también campos de direcciones de bloques repartidos en general entre dos tipos:

- direcciones que apuntan a los primeros bloques de datos del archivo;
- direcciones que apuntan a bloques que contienen otros campos de direcciones;
- y en este último caso de manera recursiva (direcciones de direcciones que apuntan ellas mismas a otras direcciones), formando un árbol, o bien cada una las hojas (terminaciones) apunta a un bloque de datos. Se habla de bloques **de indirección** (simple, doble, triple).

A continuación presentamos un ejemplo concreto de direcciones en un inodo dentro de un sistema de archivos ext2.

Un inodo ext2 contiene diez campos que apuntan cada uno a un bloque de datos, y tres campos de indirección (apuntan a direcciones).

- El primero de estos tres campos apunta, en indirección simple, a 256 direcciones de bloques de datos.
- El segundo apunta, en indirección doble, a 256 direcciones, las cuales apuntan cada una a otras 256 direcciones de bloques de datos, o sea, 256^2 bloques de datos.

- El tercer campo apunta, en triple indirección, a 256 direcciones que a su vez apuntan a 256 direcciones, que a su vez apuntan a 256 direcciones de bloques de datos, o sea 256^3bloques de datos.

Si n es el tamaño de un bloque en bytes, el tamaño máximo de un archivo es, por lo tanto **n* (10+256+256^2+256^3) bytes**.

Para un bloque de 4096 bytes, ¡eso representa unos 64 GB!

d. Tablas de catálogo

Si un inodo no contiene el nombre del archivo, éste se coloca en una tabla de catálogo. Esta tabla no es más que un directorio. Como un directorio contiene una lista de archivos y un inodo representa un archivo, entonces cada nombre de archivo se asocia a su inodo dentro del directorio.

Puede representarse esta tabla como un cuadro de dos columnas:

Tabla catálogo dir1 (directorio dir1)	
Inodo	Nombre
12345	Documento.txt
214579	Archivo.doc
47321	Música.mp3
98542	Copia.odt
...	...

e. Hard link

Un hard link permite añadir una referencia a un inodo. El hard link añade una asociación en una tabla de catálogo. No se modifican los permisos del archivo.

Un hard link no permite asignar varios nombres a un mismo directorio, ni tampoco permite efectuar vínculos desde o hacia otro sistema de archivos. Además, tenga cuidado con el contador de vínculos facilitado por el comando **ls -l**: un 1 indica que este archivo no posee otros vínculos o, dicho de otro modo, es el último. Si lo suprime, se pierde de manera definitiva. En cambio, mientras este contador sea superior a 1, si se suprime un vínculo, queda una copia del archivo en alguna parte.

```
$ touch fic1
$ ln fic1 fic2
$ ls -li
2394875 -rw-r--r-- 2 seb users    0 mar 21 22:40 fic1
2394875 -rw-r--r-- 2 seb users    0 mar 21 22:40 fic2
```

El ejemplo anterior muestra que los hard links representan el mismo archivo, ya que sólo se trata de los nombres asociados al mismo inodo. Cada uno tiene dos vínculos, lo que es lógico ya que los dos archivos apuntan al mismo inodo. Finalmente, puede ver que fic1 y fic2 tienen el mismo inodo, a saber 2394875.

2. Crear un sistema de archivos

a. mkfs, sintaxis general

Los comandos de "formateo" disponibles en Microsoft no son similares a los de Linux. Un formato de tipo Microsoft es en realidad la creación y comprobación de un sistema de archivos en una partición. La primera etapa consiste en rellenar diferentes sectores, bloques y clústeres de ceros (o de otro motivo binario) con una verificación del soporte, y la segunda consiste en la escritura de un sistema

de archivos. Esta última operación única basta para la creación de un sistema de archivos virgen en el disco o la partición.

El comando para crear un sistema de archivos es **mkfs**. **mkfs** llama a otros programas en función del tipo de sistema de archivos seleccionado.

```
mkfs -t typefs opciones periférico
```

`typefs` determina el tipo de sistema de archivos y, por lo tanto, el programa al que se llama. Existe un programa por tipo de sistema de archivos:

- **ext2**: mkfs.ext2
- **ext3**: mkfs.ext3
- **ext4**: mkfs.ext4 o mkfs.ext4dev
- **btrfs**: mkfs.btrfs
- **reiserfs**: mkfs.reiserfs
- **vfat**: mkfs.vfat (para todos los formatos FAT, pero existe mkfs.msdos)
- **ntfs**: mkfs.ntfs

Observe que es muy importante que se indiquen las opciones de cada sistema de archivos DESPUÉS de haber especificado el sistema de archivos. Si las especifica antes, serán las opciones de mkfs.

En vez de utilizar mkfs, puede utilizar directamente los programas correspondientes al tipo de sistema de archivos que se va a escribir.

b. Un primer ejemplo en ext2

Va a crear un sistema de archivos de tipo ext2 en la primera partición creada anteriormente, a saber, sdb1. El comando básico es el siguiente:

```
# mkfs -t ext2 /dev/sdb1
mke2fs 1.40.2 (12-Jul-2007)
Etiqueta de sistema de archivos=
Tipo de sistema operativo: Linux
Tamaño de bloque=4096 (log=2)
Tamaño de fragmento=4096 (log=2)
125440 i-nodos, 250472 bloques
12523 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número maximo de bloques del sistema de archivos=260046848
8 grupos de bloques
32768 bloques por grupo, 32768 fragmentos por grupo
15680 i-nodos por grupo
Superbloques de emergencia almacenados en los bloques:
        32768, 98304, 163840, 229376

Escritura de las tablas de i-nodos: completado
Escritura de los superbloques y de la información de compatibilidad del
sistema de archivos: completado
Se verificará el sistema de archivos automáticamente cada 35 montajes o
después de 180 días, según la primera eventualidad. Utilizar tune2fs -c o -i
para sobreescribir el valor.
```

Esta salida proporciona información interesante:

- Es posible poner una etiqueta (un nombre) al sistema de archivos.
- Cada bloque es de 4096 bytes.
- Hay 250472 bloques.
- Los nodos i (inodos) representan el número máximo de archivos: 125440.
- Se reserva el 5 % del espacio en disco a root, lo que significa que un usuario lambda no podrá llenar el disco a más del 95 %.
- Se reparten las tablas de inodos por grupos.
- Hay un superbloque principal y cuatro superbloques de emergencia (uno por grupo).
- Es posible modificar algunos parámetros del sistema de archivos con el comando **tune2fs**.

c. ext2, ext3 y ext4

Como los sistemas de los archivos ext2 y ext3 son compatibles, comparten los mismos parámetros, entre los cuales los más corrientes son:

Parámetro	Significado
`-b`	Tamaño de los bloques en bytes, múltiplo de 512. Si no se especifica el tamaño, se determinará por el tamaño de la partición. Cualquier archivo creado en el disco ocupa al menos un bloque y, por lo tanto, si se prevé un gran número de pequeños archivos, hay que poner un valor bajo (p. ej.: 1024).
`-c`	Verifique los bloques defectuosos antes de crear el sistema de archivos. También se puede utilizar el comando **badblocks**.
`-i`	Ratio bytes/inodo. Se calcula el tamaño de la tabla de los inodos en función del tamaño total del sistema de archivos. Un inodo ocupa 128 bytes. El hecho de meter menos limita el número de archivos posibles, pero permite ganar espacio. -i 4096: un inodo por cada 4 KB.
`-m`	Porcentaje reservado al superusuario, por defecto el 5 %. Ponerlo a cero permite ganar espacio, y root podrá trabajar en ello a pesar de todo.
`-L`	Label, etiqueta (nombre) del sistema de archivos, útil para el montaje.
`-j`	Crea un diario ext3 y por lo tanto un sistema de archivos ext3.

El ejemplo siguiente crea un sistema de archivos transaccional ext3 (opción `-j`) con un tamaño de bloques de 2048 bytes, y un inodo por cada 16 KB. Los usuarios pueden utilizar la totalidad del sistema (no se reserva ningún espacio para root). La etiqueta es DATA.

```
# mkfs -t ext2 -j -b 2048 -i 16384 -m 0 -L "DATA" /dev/sdb1
mke2fs 1.40.2 (12-Jul-2007)
Etiqueta de sistema de archivos=DATA
Tipo de sistema operativo: Linux
Tamaño de bloque=2048 (log=1)
Tamaño de fragmento=2048 (log=1)
62992 i-nodos, 500944 bloques
0 bloques (0.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de archivos=513802240
31 grupos de bloques
16384 bloques por grupo, 16384 fragmentos por grupo
2032 i-nodos por grupo
Superbloques de emergencias almacenados en los bloques:
        16384, 49152, 81920, 114688, 147456, 409600, 442368
```

```
Escritura de las tablas de i-nodos: completado
Creación del log (8192 bloques): completado
Escritura de los superbloques y de la información de compatibilidad del
sistema de archivos: completado
```

Observe que la línea de comandos siguiente tiene el mismo efecto, ya que el sistema de archivos ext3 induce el parámetro `-j`:

```
# mkfs -t ext3 -b 2048 -i 16384 -m 0 -L "DATA" /dev/sdb1
```

De ext2 a ext3

Ext3 es un sistema de archivos ext2 al cual se ha añadido un soporte transaccional. Puede convertir un sistema de archivos ext2 en ext3 utilizando **tune2fs**.

```
# tune2fs -j /dev/sdb1
tune2fs 1.40.2 (12-Jul-2007)
Creación del i-nodo del diario: completado
Se verificará el sistema de archivos automáticamente cada 25 montajes
o después de 180 días, según la primera eventualidad. Utilizar tune2fs -c
o -i para sobreescribir el valor.
```

De ext3 a ext2

Para volver a ext2, hay que suprimir otra vez el soporte transaccional con `tune2fs` y el parámetro `-O` (Option, con O mayúscula):

```
# tune2fs -O ^has_journal /dev/sdb1
```

Verifique la posible presencia de un archivo **`.journal`** y suprímalo. Finalmente, efectúe una verificación con fsck.

De ext3 a ext4

Primero, convierta su sistema de archivos a ext3 añadiendo un diario, como se ha visto anteriormente. Aunque a continuación ya se podría montar directamente este sistema ext3 como ext4, no podría beneficiarse de las ventajas de ext4 si no utilizara los extents.

El comando tune2fs permite añadirlos, así como otros parámetros necesarios:

- **extents**: añadir extents.
- **unint_bg**: optimización de la tabla de inodos que permite reducir las comprobaciones del sistema de archivos.
- **dir_index**: modificación de la tabla de catálogos para acelerar el acceso, para directorios de gran tamaño.

```
# tune2fs -O extents,unint_bg,dir_index /dev/sdb1
```

Las optimizaciones no se activarán hasta después de un primer control del sistema de archivos, lo que verá en la sección Comprobar, ajustar y arreglar. Sin embargo, debe ejecutar el comando siguiente:

```
# fsck -pDf /dev/sdb1
```

Después de esta etapa, el sistema de archivos puede montarse y usarse directamente, como se indica en la sección Acceder a los sistemas de archivos. Sin embargo, la conversión todavía no es completa, ya que los archivos que están ya en el sistema de archivos no se han optimizado para usarse con extents. El comando **chattr** permite modificar los atributos específicos de cada archivo, a

los que se les añade los extents con el parámetro +e. Necesitará listar todos los archivos y añadirles este atributo; lo podrá hacer con el comando **find**. Para empezar, monte el sistema de archivos:

```
#mount /dev/sdb1 /mnt
```

A continuación aplique el atributo `+e` en cada archivo. El parámetro `xdev` indica a find que no tiene que descender en subdirectorios que no sean del mismo sistema de archivos que el inicial:

```
# find /mnt -xdev -type f -exec chattr +e {} \ ;

# find /mnt -xdev -type d -exec chattr +e {} \ ;
```

¡Atención! Si convierte el sistema de archivos raíz o el que contiene el núcleo de Linux compruebe para empezar si GRUB es compatible con ext4. En caso contrario, corre el riesgo de no poder arrancar su sistema.

De ext4 a ext3

Si el sistema de archivos ext4 no tiene aún la opción de extents, entonces basta con montarlo a ext3. En caso contrario, no se puede realizar la conversión, ya que el atributo, una vez asignado, no debe quitarse. En este caso, la única solución consiste en realizar una copia de seguridad, para poder volver a crear el sistema de archivos ext3 y restaurar la copia.

Label

Puede visualizar y cambiar la etiqueta del sistema de archivos tecleando e2label.

```
# e2label /dev/sdb1
DATA
# e2label /dev/sdb1 OLDDATA
# e2label /dev/sdb1
OLDDATA
```

Es importante acordarse de modificar las opciones de montaje en consecuencia.

Una etiqueta de sistema de archivos no debe superar los 16 caracteres o se truncará.

d. reiserfs

Crear un sistema de archivos en `reiserfs` es tan simple como con ext2, ext3 o ext4. En el ejemplo siguiente, observe que el programa le pide una confirmación antes de crear el sistema de archivos. A diferencia de lo que se indica, no hay motivo para volver a reiniciar. Hemos suprimido el "blabla" de las primeras líneas.

```
# mkfs -t reiserfs /dev/sdb1
mkfs.reiserfs 3.6.19 (2003 www.namesys.com)
...
Guessing about desired format.. Kernel 2.6.22.17-0.1-default is running.
Format 3.6 with standard journal
Count of blocks on the device: 250464
Number of blocks consumed by mkreiserfs formatting process: 8219
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
```

```
inode generation number: 0
UUID: c1c5ac84-8ce2-4475-829d-f454b7bdb91e
CUIDADO: YOU SHOULD REBOOT AFTER FDISK!
       ALL DATA WILL BE LOST ON '/dev/sdb1'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/sdb1.
```

reiserfs acepta parámetros diferentes de ext2/3:

Parámetro	Significado
-b	Tamaño de los bloques en bytes, múltiplo de 512 (potencia de 2: 512, 1024, 2048, 4096, 8192) incluido entre 512 y 8192. Si no se especifica el tamaño, se determinará con el tamaño de la partición.
-l	Label, etiqueta (nombre) del sistema de archivos, útil para el montaje.
-f	Fuerza la ejecución del comando sin hacer preguntas, incluyendo un disco, y no una partición.
-d	Modo debug, facilita más información.

Label

Usted puede modificar una etiqueta **reiserfs** con el comando **reiserfstune**.

```
reiserfstune -l HOME /dev/hda6
```

Acuérdese de modificar las opciones de montaje correspondientes.

e. xfs

Cree un sistema de archivos xfs como éste:

```
# mkfs -t xfs -f /dev/sdb1
meta-data=/dev/sdb1              isize=256    agcount=8, agsize=31309
blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=250472, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal log           bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks, lazy-count=0
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

f. btrfs

Atención, aunque se ve cierta estabilización y es empleado a veces en sistemas de producción, btrfs sigue siendo experimental. Es posible que no haya soporte ni herramientas en su distribución, o que estén disponibles mediante la instalación de paquetes adicionales, o desactualizados.

El sistema de archivos btrfs dispone de muchas funciones avanzadas: RAID, subvolúmenes, snapshots, etc. Algunas de estas nociones serán abordadas en el último capítulo sobre el particionamiento avanzado.

La creación de un sistema de archivos btrfs sigue el mismo principio:

```
# mkfs -t btrfs /dev/sdb1
WARNING! - Btrfs Btrfs v0.19 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
fs created label (null) on /dev/sdb1
nodesize 4096 leafsize 4096 sectorsize 4096 size 401.54GB
Btrfs Btrfs v0.19
```

De ext3 o ext4 a btrfs

Los comandos btrfs permiten convertir fácilmente un sistema de archivos ext3 o ext4, con la posibilidad de volver al estado anterior si surgen problemas. En este caso, evidentemente, las modificaciones se borrarán. Para empezar, realice una comprobación del sistema de archivos:

```
# fsck -f /dev/sdb1
```

A continuación ejecute el comando **btrfs-convert**; puede tardar un poco según el tamaño de su sistema de archivos y la cantidad de datos que contenga:

```
# btrfs-convert /dev/sdb1
creating btrfs metadata.
Creating ext2fs image file.
Cleaning up system chunk.
Conversion complete.
```

Monte el sistema de archivos:

```
# mount -t btrfs /dev/sdb1 /mnt
```

En este punto de montaje encontrará una imagen del antiguo sistema de archivos en el directorio**ext2_saved** que se podrá montar si lo desea. Encontrará toda la información necesaria en la wiki de btrfs: https://btrfs.wiki.kernel.org/index.php/Main_Page

Para liberar el espacio y pasar definitivamente a btrfs sin vuelta atrás, puede borrar el directorio**ext2_saved**.

Vuelta atrás

Si btrfs no le convence, y si no ha borrado el directorio **ext2_saved**, puede anular la conversión, después de haber desmontado el sistema de archivos:

```
# btrfs-convert -r /dev/sda6
rollback complete
```

g. vfat

La creación de un sistema de archivos VFAT se hace también de la misma manera. Esta vez, va a crearlo en la partición sdb5 prevista a tal efecto. El comando va a seleccionar automáticamente, en función del tamaño de la partición, el tipo de FAT que se debe crear (12, 16 o 32). Se ha añadido el parámetro `-v` para ver las trazas de su creación.

```
# mkfs -t vfat -v /dev/sdb5
mkfs.vfat 2.11 (12 Mar 2005)
Auto-selecting FAT32 for large filesystem
/dev/sdb5 has 64 heads and 62 sectors per track,
logical sector size is 512,
using 0xf8 media descriptor, with 5856706 sectors;
file system has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 5709 sectors, and provides 730657 clusters.
```

```
Volume ID is 47df9209, no volume label.
```

Los comandos **mkfs.vfat** y **mkfs.msdos** son vínculos simbólicos hacia el programa mkdosfs.

Puede especificar varios parámetros, en particular si desea forzar la creación de un tipo de FAT dado:

Parámetro	Significado
-c	Verifica el periférico antes de la creación.
-F	Tamaño de la FAT (12, 16, 32).
-I	Permite utilizar un disco completo, y no una partición (práctico para algunos lectores de MP3).
-n	Nombre del volumen (etiqueta, label).
-v	Visualización de detalles durante la creación.

Las mtools

Las mtools son herramientas que permiten trabajar en sistemas de archivos FAT y VFAT como si estuviera bajo MSDOS o la consola de Windows. Retoman la sintaxis de los comandos de origen, pero añaden una m delante: mdir, mformat, mlabel, mdeltree, etc.

Se representan los discos y las particiones con letras de unidad c :, d :, e :. Dichas letras pueden representar un disco, una partición o un directorio. Sin embargo, debe modificar un archivo de configuración /etc/mtools.conf. Por ejemplo, para declarar /dev/sdb5 como d: añada o modifique la línea siguiente:

```
drive d: file="/dev/sdb5"
```

Es útil para modificar después ciertos datos, como el nombre del volumen del sistema de archivos vfat:

```
# mlabel -s d:
 Volume has no label
# mlabel d:
 Volume has no label
Enter the new volume label : DATAFAT
# mlabel -s d:
 Volume label is DATAFAT
```

Acceder a los sistemas de archivos

1. mount

El comando **mount** permite acceder a los periféricos de tipo bloque (las particiones) donde está el sistema de archivos. El comando **mount** sitúa el sistema de archivos que se van a montar en un punto del sistema principal llamado punto de montaje (mountpoint).

```
mount -t typefs -o opciones periférico punto_de_montaje
```

a. Montaje por periférico

Como la partición sdb1 dispone de nuevo de un sistema de archivos ext3, el comando siguiente monta la raíz del sistema de archivos que contiene sdb1 en el directorio /mnt/DATA.

```
# mount -t ext3 /dev/sdb1 /mnt/DATA
```

El comando **mount**, cuando se ejecuta sin parámetros, muestra todos los detalles en los sistemas de archivos actualmente montados (periféricos, sistema de archivos, punto de montaje, opciones):

```
# mount
/dev/sda6 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda7 on /home type ext3 (rw,acl,user_xattr)
/dev/sda1 on /windows/C type fuseblk (rw,noexec,nosuid,nodev,noatime,
allow_other,default_permissions,blksize=4096)
securityfs on /sys/kernel/security type securityfs (rw)
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

Se puede acceder a la misma información visualizando el contenido del archivo **/etc/mtab**.

Montaje por etiqueta

Es tremendamente práctico utilizar etiquetas para los sistemas de archivos montados en periféricos. Por ejemplo, cuando se procede a reordenar los discos (movimientos en los arrays de discos SCSI, por ejemplo), se modifica la ordenación de los periféricos con respecto al sistema de archivos principal. Volver a nombrar los periféricos obliga a modificar el archivo **/etc/fstab** para cada uno de los discos. Esto se evita con el uso de etiquetas. Utilice el parámetro `-L` de mount, seguido del nombre del volumen, como en el siguiente ejemplo:

```
# mount -t ext3 -L DATA /mnt/DATA
# mount
...
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

Se puede obtener la lista de las etiquetas de disco actualmente reconocidas por Linux listando el directorio `/dev/disk/by-label`. Observe que la etiqueta es un vínculo simbólico hacia el archivo periférico correspondiente:

```
# ls -l /dev/disk/by-label/
total 0
lrwxrwxrwx 1 root root 10 mar 18 14:00 DATA -> ../../sdb1
```

```
lrwxrwxrwx 1 root root 10 mar 18 14:00 DATAFAT -> ../../sdb5
```

Montaje por UUID

Cada sistema de archivos dispone de un identificador único llamado **UUID** que es un sinónimo de GUID en Linux: un número aleatorio codificado en bits lo suficientemente complejo como para que todos sean diferentes en uno o varios sistemas dados. De este modo, si el disco cambia de posición lógica, el UUID no cambia y mount encuentra el sistema de archivos, cuando en teoría es más probable que dos sistemas de archivos lleven la misma etiqueta.

Existen varios métodos para conocer el UUID de una partición. Si la herramienta **udev** se utiliza en su Linux, entonces probablemente usted disponga del comando **vol_id**. Es posible que la ruta hasta el comando no esté en la variable de rutas predefinidas PATH. Es el caso en openSUSE, que lo coloca en **/lib/udev**.

```
# ./vol_id -u /dev/sdb1
67f6e4b8-635c-4103-9a81-877fb7db29fe
```

En las distribuciones más recientes, ya no existe vol-id. El comando **blkid** lo reemplaza eficazmente:

```
# blkid /dev/sdb1
/dev/sdb1: UUID="67f6e4b8-635c-4103-9a81-877fb7db29fe"
SEC_TYPE="ext2" TYPE="ext3"
```

Para mostrar sólo el UUID, especifique únicamente este tag con el parámetro -s UUID y únicamente el valor con el parámetro -o value:

```
# blkid -o value -s UUID /dev/sdb1
67f6e4b8-635c-4103-9a81-877fb7db29fe
```

Si su sistema de archivos es de tipo ext2 o ext3, el comando **dumpe2fs** devuelve mucha información, entre la que se encuentra el UUID:

```
# dumpe2fs -h /dev/sdb1 | grep UUID
dumpe2fs 1.40.2 (12-Jul-2007)
Filesystem UUID:          67f6e4b8-635c-4103-9a81-877fb7db29fe
```

Al igual que en el caso de las etiquetas, el archivo `/dev/disk/by-uuid` contiene los enlaces simbólicos de los UUID que apuntan al archivo periférico correspondiente:

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 mar 18 14:00 47DF-9209 -> ../../sdb5
lrwxrwxrwx 1 root root 10 mar 18 14:00 67f6e4b8-635c-4103-9a81-
877fb7db29fe -> ../../sdb1
lrwxrwxrwx 1 root root 10 mar  3 09:23 a1cc2282-b6f4-46e1-bc94-
91585f1c5872 -> ../../sda5
lrwxrwxrwx 1 root root 10 mar  3 09:23 C2B072B5B072AF91 -> ../../sda1
lrwxrwxrwx 1 root root 10 mar  3 09:23 c56b96b5-e52f-453a-ba9d-
aa1df6f0c3c0 -> ../../sda7
lrwxrwxrwx 1 root root 10 mar  3 09:23 dd5e92d3-b931-4c18-91a3-
5edccc57ced9 -> ../../sda6
```

Al final, los contenidos de **/dev/disk,/by-partlable** y **/dev/disk/by-partuuid** contienen los vínculos de las etiquetas y UUID que señalan los periféricos correspondientes, para los discos en formato GPT:

```
# ls -l /dev/disk/by-partlabel/
```

```
lrwxrwxrwx. 1 root root 10 25 jun  21:09 root -> ../../sdb1
# ls -l /dev/disk/by-partuuid/
lrwxrwxrwx. 1 root root 10 25 jun  21:09 81cdf84e-d87b-48c5-a0c8-13ee07134d56
-> ../../sdb1
```

Observe que vol_id reconoce los UUID de la mayoría de los sistemas de archivos, incluyendo FAT y NTFS. Además, tanto `vol_id` como `dumpe2fs` muestran mucha más información. Por ejemplo, `vol_id -l <periférico>` devuelve la label del periférico (volumen name, etiqueta).

Para montar un sistema de archivos por UUID, utilice el parámetro `-U` de mount:

```
# mount -t ext3 -U 67f6e4b8-635c-4103-9a81-877fb7db29fe /mnt/DATA
# mount
/dev/sdb1 on /mnt/DATA type ext3 (rw)
```

Volver a montar un sistema de archivos

Si necesita modificar alguna de las opciones de montaje, no está obligado a desmontar y volver a montar después el sistema de archivos. Si modifica una opción de montaje del sistema de archivos (mediante el parámetro `-o`), puede incluir la opción remount para que se tome en cuenta en seguida la modificación. No vuelva a insertar la línea de comandos completa, sino únicamente el periférico o el punto de montaje. En el ejemplo siguiente se vuelve a montar el sistema de archivos en modo de sólo lectura:

```
# mount -o ro,remount /mnt/DATA
# mount
...
/dev/sdb1 on /mnt/DATA type ext3 (ro)
```

b. Opciones de montaje

Cada sistema de archivos acepta un cierto número de opciones de montaje que se pueden especificar después del parámetro `-o` de mount. Se separan las opciones por comas. Salvo indicación contraria, las opciones siguientes funcionan con ext2, ext3 y ext4.

Opción	Significado
`defaults`	Casi siempre presente, la opción **defaults** sustituye a las opciones rw, suid, dev, exec, auto, nouser y async por defecto.
`sync/async`	Activa o desactiva las escrituras síncronas. Con async la escritura pasa por una memoria intermedia que la difiere (más eficaz) devolviendo el control más rápido. Es preferible activar la escritura síncrona en soportes externos (pendrives, discos USB/Firewire/eSATA, etc.).
`exec/noexex`	Permite la ejecución/no ejecución de los archivos binarios en el soporte.
`noatime`	Evita la actualización de la marca temporal de acceso a un archivo (aconsejable para los soportes externos, discos SSD, páginas web, newsgroups, etc.).
`auto/noauto`	Con auto, el sistema de archivos se monta automáticamente. Con noauto, sólo se puede montar explícitamente (ver fstab).
`user/nouser`	Cualquier usuario puede montar el sistema de archivos (implica noexec, nosuid y nodev)/sólo root tiene permiso para montar el sistema de archivos (ver fstab).

remount	Remontaje del sistema de archivos para tener en cuenta nuevas opciones.
ro/rw	Montaje en modo de sólo lectura o lectura y escritura.
dev/nodev	Interpretar/No interpretar los archivos especiales.
noload	Para ext3, no carga el diario.
usrquota/ grpquota	Ignorado por el propio sistema de archivos, pero utilizado por el subsistema de cuotas.
acl	Permite el uso de los Access Control Lists.
user_xattr	Para ext3 y xfs, acepta los atributos extendidos en los archivos, por ejemplo para pegar en ellos información adicional (la codificación del texto, etc.), campos de indexación (utilizados por Beagle por ejemplo), etc.
umask	Para FAT/NTFS, aplica otra máscara global que la predeterminada (ex 133).
dmask=/fmask=	FAT/NTFS, diferencia las máscaras para los directorios y los archivos.
uid=/gid=	Como no se gestionan los permisos y propietarios, FAT/NTFS aplica un usuario o un grupo por defecto en los archivos (ej gid=users).
ssd	BTRFS, activa la optimización para los discos SSD. Las versiones más recientes de btrf aplican esta opción automáticamente si detectan un disco "no rotativo".
compress	BTRFS, activa la compresión automática de los datos. A partir del núcleo 2.6.38 puede seleccionar el formato: compress=zlib o compress=lzo.
autodefrag	BTRFS, activa la desfragmentación automática del sistema de archivos. Sólo disponible en el núcleo 3.0 y superior.

c. umount

El comando **umount** desmonta el sistema de archivos del punto de montaje.

```
# umount /mnt/DATA
```

Si se están utilizando uno o varios archivos del sistema de archivos por desmontar, **umount** no funcionará. Debe asegurarse de que ningún proceso acceda al sistema de archivos que pretende desmontar.

```
# umount /mnt/DATA
umount: /mnt/DATA: periférico ocupado
```

El comando **lsof** le ayuda a determinar qué proceso está utilizando un archivo del punto de montaje en el momento de iniciar el comando. En nuestro ejemplo, el proceso que está accediendo al archivo es el shell bash ejecutado por el usuario seb (probablemente debido a que el directorio actual es /mnt/DATA).

```
# lsof /mnt/DATA
COMMAND  PID USER   FD   TYPE DEVICE SIZE  NODE  NAME
bash    5366  seb  cwd    DIR   8,17 4096    2   /mnt/DATA
```

Como administrador, puede iniciar **fuser** para forzar la parada de los procesos que estén accediendo al punto de montaje. Es más que probable que no le guste nada al usuario correspondiente (en el caso presentado aquí, se parará su shell y se cerrará la sesión).

```
# fuser -km /mnt/DATA
```

d. /etc/fstab

El archivo **/etc/fstab** contiene una configuración estática de las diferentes opciones de montaje de los sistemas de archivos. Este archivo es invocado siempre que se inicia el sistema, ya que es aquí donde se especifican los periféricos y sus puntos de montaje. Contiene seis campos.

```
periférico punto_de_montaje typefs opciones dump fsck
```

Los espacios o las tabulaciones hacen de separadores de campos.

Campo	Descripción
periférico	El periférico que se va a montar. Se puede especificar como ruta de periférico (/dev/hda1 por ejemplo), etiqueta de sistema de archivos si existe (LABEL=/home) UUID (UUID=xxxx). Para GPT podemos emplear además PARTUUID (PARTUUID=xxxx) o PARTLABEL (PARTLABEL=root).
punto de montaje	El directorio de acceso al sistema de archivos montado.
typefs	El tipo (ext2, ext3, ext4, btrfs, reiser, vfat, etc.) del sistema de archivos.
opciones	Opciones de montaje separadas por comas.
dump	Frecuencia de volcado para las herramientas de copia de seguridad.
fsck	Frecuencia de verificación del sistema de archivos. 0=ignorar. 1=en primero, 2 en segundo, etc. Se verifican en paralelo los sistemas que tienen el mismo número.

A continuación presentamos un ejemplo (truncado, ya que los sistemas de archivos virtuales no aparecen) de archivo `/etc/fstab`:

```
/dev/sda3  /         ext4   acl,user_xattr                              1 1
/dev/sda2  /boot     ext4   acl,user_xattr                              1 2
/dev/sdb1  /home     ext4   acl,user_xattr                              1 2
/dev/sda6  /public   ext4   acl,user_xattr                              1 2
/dev/sda1  /windows  ntfs   noauto,users,gid=users,umask=0002,utf8=true 0 0
/dev/sda5  swap      swap   defaults                                    0 0
```

En vez de especificar nombres de periféricos estáticos, puede ser preferible especificar una etiqueta (label, volumen) o un UUID.

```
LABEL=BOOT   /boot                           ext4    acl,user_xattr     1 2
UUID=f0bed37c-9ddc-4764-ae7f-133205c36b5d ext3    acl,user_xattr     1 2
```

Para terminar, si no desea utilizar ninguna de estas soluciones, pero tiene que usar rutas, puede valerse de los vínculos simbólicos presentes en cada `/dev/disk/by-XXX`, donde xxx representa:

- id: los identificadores de controlador, hardware y partición de los diferentes volúmenes;
- label: los identificadores mediante etiqueta;
- uuid: los identificadores uuid;
- path: los identificadores por ruta física (bus, lun, etc.).

Aquí tiene un ejemplo:

```
./by-id:
lrwxrwxrwx 1 root root  9 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-> ../../sda
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-HDT722516DLA380_VDKD1CTCE25WSK
-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 mar 20 08:16 ata-ST3160811AS_6PT1LX2M-part1 ->
../../sdb1
...
./by-label:
lrwxrwxrwx 1 root root 10 mar 20 13:44 EXTERNE -> ../../sdc1
lrwxrwxrwx 1 root root 10 mar 20 08:16 ROOT -> ../../sda3
lrwxrwxrwx 1 root root 10 mar 20 08:16 BOOT -> ../../sda2
...
./by-path:
lrwxrwxrwx 1 root root  9 mar 20 13:44 pci-0000:00:1a.7-usb-0:2:1.0-scsi-
0:0:0:0 -> ../../sdc
lrwxrwxrwx 1 root root 10 mar 20 13:44 pci-0000:00:1a.7-usb-0:2:1.0-scsi-
0:0:0:0-part1 -> ../../sdc1
lrwxrwxrwx 1 root root  9 mar 20 08:16 pci-0000:00:1f.2-scsi-0:0:0:0 ->
../../sda
...
./by-uuid:
lrwxrwxrwx 1 root root 10 mar 20 08:16 02FCDA46FCDA339F -> ../../sda1
lrwxrwxrwx 1 root root 10 mar 20 13:44 470E-63A6 -> ../../sdc1
lrwxrwxrwx 1 root root 10 mar 20 08:16 527585d3-1e52-4aba-b7fc-70f183884
58d -> ../../sda6
...
```

Montaje durante el arranque

Durante la secuencia de inicio, uno de los scripts lee el archivo **`/etc/fstab`**, casi al principio del boot, entre la carga del núcleo y el inicio de los servicios. Todos aquellos sistemas de archivos que no tengan configurada la opción noauto se montan automáticamente porque el montaje automático está implícito. El primero es el sistema de archivos raíz /. Luego vienen la swap y los demás sistemas de archivos especificados (p. ej.: /home, /usr, etc.) para terminar con los sistemas virtuales /proc, /sys, /dev/pts, etc.

Montaje manual

Se puede utilizar el contenido de `/etc/fstab` después de la puesta en marcha del sistema para montar y desmontar de manera puntual los sistemas de archivos que no disponen, por ejemplo, de la opción **noauto**, o los soportes masivos como los lectores CD/DVD. En este caso, use el comando mount incluyendo simplemente la etiqueta, los puntos de montaje o el periférico sin tener que volver a escribir toda la línea de comandos. Mount buscará el resto de las opciones en /etc/fstab.

```
mount /home
mount -L /u01
mount LABEL=/boot
mount /dev/hda5
```

Montarlo todo

Si ha realizado modificaciones importantes en la fstab, como añadir varios puntos de montaje, puede, en vez de montar cada sistema de archivos uno por uno, montarlos todos con el parámetro `-a` de mount: `# mount -a`

bind

La opción particular bind puede ser muy práctica para montar una parte de sistema de archivos en varios puntos de montaje. Permite evitar el uso de vínculos simbólicos y sus defectos (modificación del archivo al que se señala y nombre del propio vínculo). En el ejemplo siguiente se relaciona el sistema de archivos que tiene el label **u01** con el directorio **/u01**. Luego se relaciona **/u01/applis**con el directorio **/applis**.

```
LABEL=/u01      /u01            ext3    defaults 1 2
/u01/applis     /applis         none    bind
```

e. El caso de los CD e imágenes ISO

Los CD-Rom, DVD-Rom y otros soportes de este tipo se montan como cualquier otro soporte de bloques. Los CD-Rom y algunos DVD-Rom utilizan el sistema de archivos **iso9660**.

```
# mount -t iso9660 /dev/sr0 /media/cdrom
```

La mayoría de los DVD-Rom utilizan, sin embargo, el formato **UDF** (*Universal Disk Format*).

```
# mount -t udf /dev/sr1 /media/dvd
```

Las distribuciones Linux recientes se liberan del montaje manual de los soportes externos, ya se trate de un CD, DVD, pendrive o disco externo. Los servicios udev, al conectar o insertar el soporte, se encargan de crear los archivos especiales asociados, y luego de montar y desmontar los soportes de manera automática.

Un archivo ISO es una imagen del contenido de un CD o DVD. Es un sistema de archivos iso9660 o udf en un archivo. Es posible utilizar esta imagen como periférico con ayuda de la opción loop. Esta técnica consiste en relacionar la imagen con un periférico en modo loopback, y conseguir así que las herramientas lo traten como un disco más.

```
# mount -o loop -t iso9660 image.iso /mnt/iso
```

Controlar el sistema de archivos

1. Estadísticas de ocupación

a. Por sistema de archivos

El comando **df** permite obtener estadísticas de ocupación de cada sistema de archivos montado. Sin argumento, **df** facilita información sobre todos los sistemas de archivos. Puede pasar como argumento un periférico montado o un punto de montaje. Si pasa un directorio cualquiera, df da la información del sistema de archivos que contenga este directorio.

```
# df
Sis. de fich.      1K-bloques   Ocupado Disponible Capacidad Montado en
/dev/sda3           41286828   6482952  32706592  17%       /
udev                 1031240       124   1031116   1%       /dev
/dev/sda2             521780     27092    468184   6%       /boot
/dev/sdb1          153834852  49189572  96830864  34%       /home
/dev/sda6           73142560  19150372  50276760  28%       /public
/dev/sdc1          292890560 175894672 116995888  61%       /media/EXTERNO
```

El resultado es explícito. La unidad por defecto es el KB (idéntico al parámetro `-k`) aunque la norma POSIX define una unidad de bloque en 512 bytes. Puede modificar los parámetros para pedir el resultado en MB (`-m`).

Para que sea más legible, añada el parámetro `-h` (*Human readable*).

No confunda este último parámetro con -H, que visualiza el resultado en unidades **SI** (*Sistema Internacional*).

```
# df -m /home
Sis. de fich.      1M-bloques  Ocupado  Disponible Capacidad Montado en
/dev/sdb1              150230   48043       94557    34%      /home
```

```
# df -h /home
Sis. de fich.      Tam.   Oc.  Disp. %Oc.  Montado en
/dev/sdb1          147G   47G   93G  34%   /home
```

```
# df -H /home
Sis. de fich.      Tam.   Oc.  Disp.  %Oc.  Montado en
/dev/sdb1          158G   51G  100G   34%   /home
```

Las unidades del SI que definen las unidades de peso y medida se basan en potencias de 10. Es sencillo memorizar que 1 Kg es igual a 10^3 gramos, o sea, 1000 gramos. Por lo tanto, 1 KB equivale a 10^3 bytes, o sea 1000 bytes... ¿No está de acuerdo? Un ordenador no trabaja con potencias de 10, sino de 2. A nivel físico, 1 KB equivale 2^{10} bytes, o sea 1024 bytes; 1 MB vale 2^{20} bytes, y así sucesivamente. Este método se llama método tradicional. El sistema internacional prefiere emplear los términos kibibyte (kilo Binario, Kib), mebibytes (Meb) y gibibytes (Gib) para las representaciones binarias, y KB, MB y GB para las potencias de 10, como en el caso de los metros y los gramos. Ahora entiende por qué un disco de 160 GB se corresponde en realidad a 152,5 Gib. En cierto modo, nos dejamos engañar de manera legal y oficial.

La `-T` añade la visualización del tipo de sistema de archivos.

```
# df -T /home
Sis. de fich. Tipo   1K-bloques  Ocupado Disponible  Capacidad  Montado en
/dev/sdb1     ext3   153834852  49197688  96822748   34%        /home
```

El comando **df** permite también facilitar estadísticas para el uso de los inodos. Puede combinar los parámetros `-i` y `-h`.

```
# df -i /home
Sis. de fich.         Inodos   IUtil.  ILib.   %IUti. Montado en
/dev/sdb1            19546112  86016  19460096    1%  /home
# df -ih /home
Sis. de fich.         Inodos   IUtil.  ILib.   %IUti. Montado en
/dev/sdb1               19M     84K    19M       1%  /home
```

b. Por estructura

El comando **du** (*disk usage*) facilita la información relativa al espacio ocupado por una estructura (un directorio y todo su contenido). Si no se especifica nada, se utiliza el directorio corriente. Los parámetros `-k` (Kb) y `-m` (MB) determinan la unidad. Se facilita el tamaño para cada elemento (incluso redondeado). El tamaño total de la estructura está en la última línea.

```
# du -m LIBRO_ALGO
1       LIBRO_ALGO/BACKUP/capítulo7/codigo_java
2       LIBRO_ALGO/BACKUP/capítulo7
1       LIBRO_ALGO/BACKUP/Introducción
1       LIBRO_ALGO/BACKUP/capítulo4/ilustraciones
...
42      LIBRO_ALGO/
```

Para obtener el total, y no todos los detalles, utilice `-s`.

Observe que no se limita **du** a un único sistema de archivos y sigue calculando si encuentra un punto de montaje en la estructura que analiza. Si quiere limitar el cálculo al sistema de archivos corriente sin entrar en los puntos de montaje presentes en la estructura, especifique `-x`.

```
# du -ks LIBRO_ALGO
42696 LIBRO_ALGO/
```

```
# du -msx /
1064 /
```

2. Comprobar, ajustar y arreglar

a. fsck

El comando **fsck** permite comprobar y arreglar un sistema de archivos.

```
fsck -t typefs periférico
```

El sistema de archivos que se quiere comprobar o arreglar no debería estar montado, o, como mucho, montado en modo de sólo lectura.

De la misma forma que **mkfs**, **fsck** invoca a otro comando teniendo en cuenta el tipo del sistema de archivos para comprobar: esos otros comandos más

especializados son fsck.ext2, fsck.ext3, etc. Cada uno puede presentar opciones particulares. Si **fsck** no reconoce la opción que se le proporciona, la transmite al programa correspondiente. Si no indica un tipo, **fsck** intenta determinarlo por sí mismo.

Para este ejemplo, se pasa el parámetro `-f` a fsck para forzar la comprobación (no ha sido posible producir una corrupción), así como el parámetro `-V` para facilitar todos los detalles.

```
# fsck -fV /dev/sda2
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
Paso 1: verificación de los i-nodos, de los bloques y de los tamaños
Paso 2: verificación de la estructura de los directorios
Paso 3: verificación de la conectividad de los directorios
Paso 4: verificación de los contadores de referencia
Paso 5: verificación de la información del sumario del grupo

     42 inodes used (0.06%)
      1 non-contiguous inode (2.4%)
        # of inodes with ind/dind/tind blocks: 10/1/0
   8864 blocks used (6.69%)
      0 bad blocks
      1 large file

     27 regular files
      3 directories
      0 character device files
      0 block device files
      0 fifos
      0 links
      3 symbolic links (3 fast symbolic links)
      0 sockets
--------
     33 files
```

Cuando el sistema de archivos está dañado, **fsck** inicia una batería de preguntas por cada acción necesaria. Puede pasar el parámetro `-p` para intentar una reparación automática, o también `-y`para forzar las respuestas a sí.

Durante el inicio del sistema, éste comprueba desde hace cuánto tiempo, o después de cuántos montajes, no se ha comprobado el sistema de archivos. Si el intervalo de tiempo es demasiado grande, ejecutará un **fsck** en el sistema de archivos correspondientes. Se pueden modificar los intervalos mediante el comando **tune2fs**.

Una partición de tipo btrfs no necesita fsck, emplea **btrfsck** o **btrfs check**. El comando fsck.btrfs sólo existe para indicar el uso de los dos primeros.

b. badblocks

El comando **badblocks** intenta comprobar los bloques defectuosos en el periférico de almacenamiento proporcionado como argumento. **mkfs** o **fsck** pueden llamar a este comando si se les proporciona el parámetro `-c` (check).

Por defecto, **badblocks** lee la totalidad de los bloques del soporte y devuelve un error si uno o varios de ellos son ilegibles. Se puede ejecutar el comando incluso aunque el sistema de archivos esté montado, excepto si usted una prueba en lectura y escritura, incluso no destructiva.

```
# badblocks -v /dev/sda2
Comprobación de los bloques 0 a 530144
Comprobación de los bloques defectuosos (prueba en modo de sólo lectura): done
Paso completado, 0 bloques defectuosos localizados.
```

Los parámetros `-n` (no destructivo) y `-w` (write, con motivos, destructivo) intentan escribir en los bloques. El primero lee y vuelve a escribir lo leído en el bloque, el segundo escribe otro tipo de información (0xaa, 0x55, 0xff, 0x00) y por lo tanto sobreescribe lo anterior.

La ejecución de badblocks puede ser larga, varias horas en algunos centenares de GB.

c. dumpe2fs

El comando **dumpe2fs** acepta como argumento un periférico que contiene un sistema de archivos ext2, ext3 o ext4. Devuelve un gran número de información sobre el sistema de archivos.

```
# dumpe2fs /dev/sda2
dumpe2fs 1.40.2 (12-Jul-2007)
Filesystem volume name:   /boot
Last mounted on:          <not available>
Filesystem UUID:          abc32a5a-a128-4492-8e03-248521015835
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal resize_inode dir_index filetype
needs_recovery sparse_super large_file
Filesystem flags:         signed directory hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              66400
Block count:              132536
Reserved block count:     6626
Free blocks:              123672
Free inodes:              66358
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      32
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         13280
Inode blocks per group:   415
Filesystem created:       Sat Feb 23 22:52:05 2008
Last mount time:          Thu Mar 20 19:13:51 2008
Last write time:          Thu Mar 20 19:13:51 2008
Mount count:              1
Maximum mount count:      500
Last checked:             Thu Mar 20 19:07:48 2008
Check interval:           5184000 (2 months)
Next check after:         Mon May 19 20:07:48 2008
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal inode:            8
Default directory hash:   tea
Directory Hash Seed:      f1584155-5760-4445-8009-0444ffa81f91
Journal backup:           inode blocks
Tamaño del journal:       16M

Grupo 0: (Bloques 0-32767)
  superbloque Primario a 0, Descriptores de grupos a 1-1
```

Esta salida es muy larga (y ello a pesar de que ha sido truncada), pero le da todos los detalles posibles sobre el sistema de archivos. Puede aislar únicamente el encabezamiento (hasta el tamaño del diario) con el parámetro `-h`. Si busca un dato en concreto, lo mejor es utilizar el comando **grep**.

```
  Bloques reservados GDT a 2-33
  Bitmap de bloques a 34 (+34), Bitmap de i-nodos a 35 (+35)
  Tabla de i-nodos a 36-450 (+36)
  25696 bloques libres, 13256 i-nodos libres, 2 directorios
  Bloques libres: 4559-5660, 6767-12287, 12289-14335, 15644-26721, 26820-32767
  I-    nodos libres: 14, 16-20, 23-24, 33-13280
...
```

```
# dumpe2fs -h /dev/sda2|grep -i "block size"
dumpe2fs 1.40.2 (12-Jul-2007)
Block size:               4096
```

d. tune2fs

El comando **tune2fs** permite modificar algunos parámetros de un sistema de archivos ext2 o ext3. Hemos visto anteriormente este comando cuando se explicó cómo convertir ext2 a ext3, y viceversa. Le presentamos algunos parámetros soportados por el comando:

```
# tune2fs -m 0 -s 1 -U random -e remount-ro -c 60 -i 180 ,
```

Parámetro	Significado
`-c n`	Número de veces que se debe montar el sistema de archivos antes de ser comprobado automáticamente. Por ejemplo, si n vale 10, se iniciará de manera automática fsck la décima vez que se monte. Si n vale 0 o -1, se desactivará la verificación.
`-i n`	Intervalo de tiempo entre dos comprobaciones. La unidad por defecto es el día. Se pueden juntar los sufijos d (días), w (semanas) o m (mes) al número. `-i` 180d significa que se controlará el sistema de archivos pasados 180 días.
`-j`	Añade un diario sobre un sistema de archivos ext2. Es preferible hacerlo con el sistema desmontado. En caso contrario, se añade un archivo oculto `.journal` a la raíz del sistema, inmutable (salvo si destruye el diario), que se integrará dentro del sistema durante la próxima ejecución de fsck.
`-L`	Modifica la etiqueta (label, nombre de volumen). La etiqueta no debe superar 16 caracteres o se truncará.
`-e err`	Indica cómo debe reaccionar el núcleo si se detecta un error en el sistema de archivos durante el boot. El valor por defecto es "continuo". Los demás valores posibles son "panic" (bloqueo del núcleo en modo kernel panic) y "remount-ro": remontaje en modo de sólo lectura.
`-m n`	El porcentaje n representa el tamaño reservado a los procesos iniciados por root (y el propio root) sobre el tamaño total de la partición. En un sistema de archivos reservados a los usuarios, poner 0 permite llenar el sistema de archivos hasta el 100 %. Pero es importante reservar en la raíz, o /var, una zona para que algunas bitácoras como syslogd puedan seguir escribiendo los registros. Por defecto se reserva el 5 %.
`-o [^]option`	Añade o suprime (con ^) la opción indicada por defecto al montar. Las opciones pueden ser, por ejemplo, acl o user_xattr.
`-O [^]function`	Añade o suprime (con ^) la función indicada. La función más famosa es "has_journal". `-O has_journal` equivale a -j. `-O ^has_journal` convierte ext3 en ext2.
`-U UUID`	Modifica el valor del UUID a su conveniencia (formato hexadecimal). Es posible suprimirlo (clear), generar uno de manera aleatoria (random) o generar uno en función de la fecha (time).
`-s 0/1`	Activa o desactiva la "sparse super feature". En discos de gran tamaño, la activación reduce el número de bloques de emergencia para ganar espacio. Después debe ejecutar fsck.

Observe que `-c` e `-i` van de la mano. A vencimiento del plazo se efectúa la comprobación con fsck. Luego el sistema pone de nuevo a cero la fechas y los contadores. Como consecuencia de un fsck, el sistema pone de nuevo a cero las fechas y los contadores. Se verifican los contadores durante el montaje en el momento del inicio (boot) del sistema. Si hace 300 días que el sistema no se ha reiniciado y no se ha verificado el sistema de archivos durante este intervalo, no se verificará automáticamente durante estos 300 días, sino en el próximo reinicio. Como algunos servidores reinician pocas veces (por ejemplo, cada dos años), no se comprueba el sistema de archivos de manera automática durante todo este tiempo...

La swap

1. ¿Porqué crear una swap?

En un entorno de 32 bits, un proceso puede acceder teóricamente a 4 GB de espacio de memoria. Dispone de 4 GB de memoria virtual para él de forma exclusiva. Hay varios límites a esta posibilidad:

- El espacio de memoria direccionable de un proceso se comparte entre la zona de código y la zona de datos, cuyo tamaño puede variar según el núcleo utilizado.
- No todos los ordenadores disponen de 4 GB de memoria (aunque sea habitual encontrar servidores de Linux que disponen de 16, 32, 64 GB o incluso más memoria).
- Todos los procesos deben compartir la memoria del ordenador.

En un sistema de 64 bits no hay límite de 4 GB, pero el tamaño de la memoria física disponible sigue siendo fijo.

¿Qué ocurre si un proceso no tiene bastante memoria para tratar sus datos? El sistema operativo descargará segmentos de la memoria física en una zona de intercambio en disco que hará de memoria virtual intermedia. Por lo tanto, hay un intercambio entre la memoria física y esta zona de intercambio, llamada espacio swap. Este proceso permite utilizar más memoria de la que dispone realmente el ordenador, a costa de una importante ralentización del programa si éste resulta ser muy "glotón".

2. Tamaño óptimo

No hay reglas estrictas para establecer el tamaño de la swap. Sin embargo las reglas corrientes siguientes son válidas en la mayoría de casos:

- Si la RAM tiene menos de 512 MB, la swap debe ser el doble de grande.
- Si la RAM tiene entre 1 y 4 GB, la swap debe tener el tamaño de la RAM.
- Si la RAM supera los 4 GB, la swap debe tener 4 GB más o menos, según el uso de los procesos.

Algunos servidores requieren bastante más espacio, como por ejemplo swaps de 8 o 16 GB, o incluso más. En este caso, conviene recurrir a otros modelos de arquitectura hardware o software.

3. Crear una partición de swap

- Ya sabe crear una partición. Cree una partición con fdisk del tamaño deseado para la swap y asígnele el tipo **82**.
- Sincronice la tabla de las particiones con **partprobe**.
- Utilice el comando **mkswap** para preparar la partición que se va a recibir de la swap.

```
# mkswap /dev/sda5
Inicialización de la versión del espacio de swap 1, tamaño = 2154983 kB
ninguna etiqueta, UUID=c84714e6-c42c-44d4-9fe7-10dc6afac644
```

- Su swap está lista.

Se puede asignar una etiqueta a la partición de swap con el parámetro `-L`.

Si crea más de 1 o 2 GB de swap, debería pensar en crear varias particiones de swap en discos diferentes ubicados en varios controladores físicos. Linux utilizará cada una de estas

particiones, lo que asegura accesos más rápidos.

4. Activar y desactivar la swap

a. Activación dinámica

Linux permite activar y desactivar la swap, o partes de la swap, directamente sin tener que volver a iniciar el sistema.

El comando **swapon** permite activar una partición de swap:

```
# swapon /dev/sda5
```

El parámetro `-p` permite modificar la prioridad de la swap. Cuanto más elevado sea el valor, entre 0 y 32767, más alta será la prioridad de una zona de swap. Linux usa un sistema de prioridades. Este parámetro es útil cuando maneja varias particiones de swap en diferentes discos. En tal caso, dé prioridad al disco más rápido. Si las particiones de swap tienen asignadas las mismas prioridades, Linux repartirá la carga por igual.

Como con mount, el parámetro `-L` permite activar una zona de swap gracias a su etiqueta.

El comando **swapoff** desactiva una zona de swap. Tenga cuidado en disponer del espacio de memoria libre necesario; si no, el comando no funcionará.

El contenido de **/proc/swaps** refleja el estado actual de las zonas de swap activas.

```
# cat /proc/swaps
Filename                        Type            Size    Used    Priority
/dev/sda5                       partition       1461872 2012    -1
```

b. En /etc/fstab

Las zonas de swap se colocan en el archivo **/etc/fstab**. Aquí tiene un ejemplo:

```
/dev/sda5 swap swap defaults 0 0
```

Se pueden especificar las opciones **noauto** y **pri=X**. La opción **pri** permite definir la prioridad de la zona de swap.

En el momento de iniciarse, el sistema ejecuta **swapon -a**, que activa todas las particiones de swap presentes en la fstab, excepto si se especifica noauto.

En el momento de la parada, el sistema ejecuta **swapoff -a**, que desactiva completamente la swap.

5. En caso de emergencia: archivo de swap

¿Qué hacer si le falta espacio de swap y ya no es posible crear una nueva partición? Linux sabe utilizar un archivo de intercambio (como Windows, incluso el de Windows). Si queda espacio en uno de sus sistemas de archivos, puede crear encima un archivo de intercambio con un tamaño predefinido. Esta swap será menos eficaz que una partición de swap (problema de fragmentación, tiempo de acceso al sistema de archivos).

Aquí tenemos las operaciones para una pequeña swap de 32 MB:

```
# free | grep Swap
Swap:      2104472       4344    2100128
# dd if=/dev/zero of=/swap bs=1024 count=32768
```

```
32768+0 grabaciones leídas
32768+0 grabaciones escritas
33554432 bytes (34 MB) copied, 0,35697 s, 94,0 MB/s
slyserver:~ # mkswap /swap
Inicialización de la versión del espacio de swap 1, tamaño = 33550 kB
ninguna etiqueta, UUID=b2e5e99e-09a1-4b2d-ac76-59f76526453a
slyserver:~ # chmod 600 /swap
slyserver:~ # sync
slyserver:~ # swapon -v /swap
swapon sur /swap
slyserver:~ # free | grep Swap
Swap:      2137232        4308    2132924
```

Modifique si es preciso el archivo **/etc/fstab** para activar esta swap después del montaje de los sistemas de archivos. Se activa la swap en el boot, en general después del montaje de /. Si la nueva swap se encuentra en otra parte (otro punto de montaje), obtendrá un mensaje de error porque la swap ha sido activada antes que los otros puntos de montaje. Por lo tanto, es preferible crear el archivo en el sistema de archivos raíz /.

```
/swap swap swap defaults 0 0
```

6. Estado de la memoria

a. free

El comando **free** le proporciona la información relativa a la memoria física (RAM) de su ordenador, así como a la ocupación de la swap. La unidad por defecto es el KB (idéntico con el parámetro `-k`), pero se puede cambiar a MB (`-m`) e incluso a GB (`-g`).

```
# free -k
             total       used       free     shared   buffers    cached
Mem:       2062484    2045732      16752          0    707512    776528
-/+ buffers/cache:     561692    1500792
Swap:      2104472        132    2104340
```

Sin embargo, tenga cuidado en interpretar correctamente las columnas. En el resultado anterior, aun disponiendo de 2 GB de memoria, el sistema indica que sólo hay libres 16 MB. Esto se debe a que Linux tiende a reservarse todo el espacio disponible en forma de buffers (memoria intermedia) y de caché. El contenido de la caché es volátil; por lo tanto Linux puede liberar en gran medida este espacio para asignarlo a los programas y datos. También es el caso para los buffers. Cuando se insertó el comando anterior, el ordenador estaba copiando una pista de DVD en el disco duro. Unos instantes después de finalizar la copia, el resultado es el siguiente:

```
             total       used       free     shared   buffers    cached
Mem:       2062484    1586772     475712          0      9708    996000
-/+ buffers/cache:     581064    1481420
Swap:      2104472         44    2104428
```

Se han liberado los buffers, una parte ha sido devuelta a la caché justo al principio de la codificación de la pista en DivX. Por lo tanto, durante sus cálculos de memoria libre, tenga en cuenta el hecho de que se pueden liberar las cachés. El sistema dispone de unos 1,4 GB de espacio de memoria (sumando la libre y la cacheada) que Linux puede asignar a los programas.

El parámetro **-m** muestra el resultado en MB, mientras que **-g** lo muestra en GB. El parámetro **-t**muestra el total por cada columna.

Highmem, lowmem

Si trabaja en un entorno de 32 bits, un parámetro muy interesante es `-l`, que muestra el detalle de la ocupación de memoria de zonas de memoria alta (highmem) y baja (lowmem). En estos sistemas, la memoria se descompone en las zonas baja y alta. La zona highmem es la zona en la que los procesos tienen su código y sus datos. La zona lowmem es una zona reservada al núcleo y que sirve, entre otras funciones, para almacenar diversas cachés y buffers, así como las tablas de asignación de memoria, para cada proceso. Esta memoria puede asignarse también a los procesos del núcleo. Si un proceso reserva memoria, los datos relativos a su posición y su tamaño reales están ìen lowmem. De igual modo, todas las transferencias de datos (entrada y salida de disco, por ejemplo) sólo las puede hacer el núcleo y, por lo tanto, estos datos deben transitar de la zona highmem a la zona lowmem.

Esta diferenciación puede conducir a casos sorprendentes donde el sistema operativo puede necesitar memoria y desencadenar mecanismos de autodefensa (matando procesos) mientras que la memoria no parece saturada y el swap no está en uso. El comando free muestra por defecto la suma de ambas zonas de memoria. Lo que sucede realmente es que la zona lowmem está llena. El siguiente capítulo le mostrará cómo reducir los riesgos de que esto se produzca.

```
$ free -l
             total       used       free     shared    buffers     cached
Mem:       1025488     735324     290164          0      32868     419224
Low:        886408     596528     289880
High:       139080     138796        284
-/+ buffers/cache:     283232     742256
Swap:      1999868          0    1999868
```

Los sistemas en 64 bits no tienen este problema, ya que pueden acceder a toda la memoria, que es de tipo lowmem.

b. Memoria reservada

2 GB corresponden a 2097152 KB. Ahora bien: el total presenta una diferencia de unos 34 MB. Se reserva esta memoria para el núcleo del sistema y el resto de los binarios no la pueden utilizar. Es útil para los tratamientos del núcleo, su carga, el initrd. Vea el resultado del comando siguiente (truncado de manera voluntaria):

```
# dmesg | grep -i memory
...Memory: 2057756k/2096640k available (2053k kernel code, 38496k
reserved, 1017k data, 316k init)
Freeing initrd memory: 4411k freed
Freeing unused kernel memory: 316k freed
...
```

El sistema se reserva unos 38 MB de memoria, luego libera la memoria que ya no necesita para obtener el resultado esperado.

c. meminfo

El sistema de archivos virtual /proc contiene información detallada sobre la memoria mediante el seudoarchivo **`/proc/meminfo`**. Parece bastante difícil encontrar algo más completo. La salida siguiente es el resultado en un sistema Linux de 64 bits. En 32 bits, dos líneas adicionales (highmem y lowmem) indican las zonas reservadas a los datos y al núcleo. Las primeras líneas y las relativas a la swap son idénticas al resultado del comando **free**.

```
# cat /proc/meminfo
MemTotal:          2062484 kB
MemFree:             16452 kB
Buffers:              4152 kB
Cached:            1456476 kB
SwapCached:              0 kB
```

```
Active:            1307912 kB
Inactive:           614356 kB
SwapTotal:         2104472 kB
SwapFree:          2104428 kB
Dirty:              188324 kB
Writeback:               0 kB
AnonPages:          461688 kB
Mapped:             120268 kB
Slab:                62476 kB
SReclaimable:        38384 kB
SUnreclaim:          24092 kB
PageTables:          15344 kB
NFS_Unstable:            0 kB
Bounce:                  0 kB
CommitLimit:       3135712 kB
Committed_AS:       845496 kB
VmallocTotal:  34359738367 kB
VmallocUsed:         56016 kB
VmallocChunk:  34359678971 kB
HugePages_Total:         0
HugePages_Free:          0
HugePages_Rsvd:          0
Hugepagesize:         2048 kB
```

Las cuotas de disco

1. Definiciones

Las **cuotas** permiten poner límites al uso de espacio en los sistemas de archivos. Estos límites son de dos tipos:

- **inodos**: limita el número de archivos.
- **bloques**: limita número de bloques.

Se implementan las cuotas por sistema de archivos individual, y no para el conjunto de los sistemas de archivos. Se puede gestionar cada usuario de manera totalmente independiente. Ocurre lo mismo con los grupos. Para cada utilización (inodo o bloque), puede instalar dos límites en el tiempo:

- **Límite duro** (hard): cantidad máxima de inodos o bloques utilizados que el usuario o el grupo no podrá superar en ningún caso. Este límite es taxativo (no podrá crear un archivo cuyo tamaño supera el límite).
- **Límite suave** (soft): cantidad máxima de inodos o bloques utilizados que el usuario o el grupo pueden superar de manera temporal. En este caso, las creaciones y modificaciones serán posibles hasta cierto punto marcado por límite duro o el período de gracia.
- **Período de gracia**. Durante este tiempo el usuario puede continuar trabajando en el sistema de archivos. El objetivo es que regrese al límite suave en dicho plazo. Una vez superado el período de gracia, el límite suave se convierte en el límite duro. Pase lo que pase, el usuario no podrá nunca superar el límite duro.

Las cuotas se establecen en el núcleo y en los sistemas de archivos. Para utilizarlas, se deben instalar las herramientas de cuotas (paquetes de cuota). En el ejemplo siguiente se muestra cómo instalar las herramientas en un sistema de archivos ext3.

Las cuotas están soportadas por un sistema de archivos btrfs pero no por los comandos presentados aquí. Puede dirigirse a https://btrfs.wiki.kernel.org/index.php/Quota_support para estudiar el funcionamiento de las cuotas en btrfs, diferente de este, clasico presentado aquí. No hay por ejemplo cuotas para el usuario. Sin embargo, podemos crear un subvolumen por carpeta personal y aplicar las cuotas.

2. Implementación

La implementación de las cuotas no funciona en los sistemas de archivos ext o reiserfs. Los ejemplos siguientes se basan en ext. Va a instalar los cuotas en la partición /home conforme a las etapas siguientes:

- Modifique las opciones de partición en **/etc/fstab**. Añada las opciones usrquota (usuario) o grpquota (grupo), o las dos.

```
LABEL=/home /home ext3 defaults,usrquota 1 2
```

- Monte de nuevo el sistema de archivos.

```
# mount -o remount /home
```

- Cree los archivos que contienen la información de cuota (base de datos de cuotas).

```
# cd /home
# touch aquota.user aquota.group
```

- Actualice la base de datos con el comando **quotacheck**.

```
# quotacheck -c /home
```

- Inicie (o detenga) las cuotas. Esta operación no es necesaria después de un nuevo inicio de Linux, ya que la implementación de cuotas está incluida en los scripts de inicio. El comando**quotaon** inicia las cuotas para el sistema de archivos indicado (`-a` para todos). El comando**quotaoff** deshabilita las cuotas.

```
quotaon /home
```

- Edite las cuotas para los usuarios o los grupos. Se utiliza el comando **edquota**. En la práctica, si va a establecer las mismas cuotas a todos los usuarios, se crea un usuario lambda cuyas propiedades se copiarán para los demás.

Establecer las cuotas para josé:

```
# edquota josé # = edquota -u josé
```

Las cuotas de arturo son idénticas a las de josé:

```
# edquota -p josé arturo
```

- Establezca el período de gracia. El plazo acepta las unidades "secondsé", "minutes", "hours", "days", "weeks", "months".

```
# edquota -t
```

- Verifique las cuotas. Los usuarios pueden comprobar el estado de sus cuotas con el comando**quota**. El administrador puede generar un informe con **repquota**. Finalmente, el comando**warnquota**, que puede ser programado con el cron, puede mandar un mail a los usuarios para avisarles en caso de superación.

La edición de las cuotas se hace con el editor por defecto del sistema, que suele ser vi (se puede modificar el comportamiento vía las variables EDITOR y VISUAL). Los bloques de cuotas son bloques de 1KB.

```
# edquota josé
Disk quotas for user josé (uid 502):
Filesystem  blocks    soft       hard       inodes    soft     hard
/dev/hda5   1695256   2500000    3000000    12345     0        0
```

Con el editor se pueden modificar los valores soft y hard que corresponden a los límites suaves y duros para el número de bloques y el número de inodos. Más arriba, se ha establecido un límite suave de unos 2,4 GB (2500000 KB) y duro de unos 2,9 GB (3000000 KB) de ocupación del sistema de archivos para josé. No hay cuotas en el número de inodos (valor a 0).

El contenido de los campos blocks e inodos es dinámico y no se debe tocar, ya que de todas maneras no surte ningún efecto.

```
# edquota -t
Filesystem            Block grace period     Inode grace period
/dev/hda3                     7days                  7days

# repquota /home
*** Report for user quotas on device /dev/hda5
Block grace time: 7 days; Inode grace time: 7 days
                        Block limits                File limits
User           Used     soft      hard    grace    used    soft    hard
---------------------------------------------------------------------
root      --   12345        0         0                5       0       0
roger     -- 1695256  2500000   3000000            12345       0       0
```

Finalmente, se hace necesario utilizar de forma regular el comando **quotacheck** para mantener la coherencia de la información de cuotas de los sistemas de archivos. Efectivamente, al deshabilitar las cuotas o detener el sistema de forma abrupta, resulta necesario verificar y volver a actualizar la información.

```
# quotacheck -avug
```

Los permisos de acceso

1. Los permisos básicos

a. Permisos y usuarios

El papel de un sistema operativo es también el de asegurar la integridad y el acceso a los datos, lo que es posible gracias a un sistema de permisos. A cada archivo o directorio se le asignan unos privilegios que le son propios, así como autorizaciones de acceso individuales. Al intentar acceder, el sistema comprueba si está autorizado.

Cuando el administrador crea un usuario, le asigna un **UID** (*User Identification*) único. Los usuarios quedan definidos en el archivo **`/etc/passwd`**. Del mismo modo, cada usuario se integra en, al menos, un grupo (grupo primario). Todos éstos tienen un identificador único, el **GID** (*Group Identification*) y están definidos en el archivo **`/etc/group`**.

El comando **id** permite obtener esta información. A nivel interno, el sistema trabaja únicamente con los UID y GID, y no con los propios nombres.

```
$ id
uid=1000(seb) gid=100(users) grupos=7(lp),16(dialout),33(video),
100(users)
```

Se asocian un UID y un GID a cada archivo (inodo) que define su propietario y su grupo con privilegios. Usted asigna permisos al propietario, al grupo con privilegios y al resto de la gente. Se distinguen tres casos:

- UID del usuario idéntico al UID definido para el archivo. Este usuario es propietario del archivo.
- Los UID son diferentes: el sistema comprueba si el GID del usuario es idéntico al GID del archivo. Si es el caso, el usuario pertenece al grupo con privilegios del archivo.
- En los otros casos (ninguna correspondencia): se trata del resto de la gente (others), ni es el propietario, ni un miembro del grupo con privilegios.

d	rwxr-xr-x	29	seb	users	4096	Mar 15 22:13	Documentos

En esta línea de la tabla, el directorio Documentos pertenece al usuario seb y al grupo users, y posee los permisos rwxr-xr-x.

b. Significado

Permiso	Significado
General	
r	Readable (lectura).
w	Writable (escritura).
x	Executable (ejecutable como programa).
Archivo normal	
r	Se puede leer el contenido del archivo, cargarlo en memoria, listarlo y copiarlo.
w	Se puede modificar el contenido del archivo. Se puede escribir dentro. Modificar el contenido no significa poder eliminar el archivo (ver permisos

	en directorio).
x	Se puede ejecutar el archivo desde la línea de comandos si se trata de un programa binario (compilado) o de un script (shell, perl...).
Directorio	
r	Se pueden listar (leer) los elementos del directorio (catálogo). Sin esta autorización, ls y los criterios de filtro en el directorio y su contenido no serían posibles. No obstante, puede seguir accediendo a un archivo si conoce su ruta de acceso.
w	Se pueden modificar los elementos del directorio (catálogo), y es posible crear, volver a nombrar y suprimir archivos en este directorio.**Es este permiso el que controla el permiso de eliminación de un archivo**.
x	Se puede acceder al catálogo por CD y se puede listar. Sin esta autorización, es imposible acceder al directorio y actuar en su contenido, que pasa a estar cerrado.

Así, para un archivo:

rwx	r-x	r--
Permisos para el propietario de lectura, escritura y ejecución.	Permiso para los miembros del grupo de lectura y ejecución.	Permisos para el resto del mundo de lectura únicamente.

2. Modificación de los permisos

Cuando se crea, un archivo o un directorio dispone de permisos por defecto. Utilice el comando**chmod** (*change mode*) para modificar los permisos en un archivo o un directorio. Existen dos métodos para modificar estos permisos: mediante símbolos o mediante un sistema octal de representación de permisos. Sólo el propietario de un archivo puede modificar sus permisos (además del administrador del sistema). El parámetro `-R` cambia los permisos de manera recursiva.

a. Mediante símbolos

La sintaxis es la siguiente:

```
chmod modificaciones Fic1 [Fic2...]
```

Si hay que modificar los permisos del propietario, utilice el carácter **u**; para los permisos del grupo con permisos, el carácter **g**; para el resto, el carácter **o**, y para todos, el carácter **a**.

Para añadir permisos, se utiliza el carácter **+**; para retirarlos, el carácter **-**, y para no tener en cuenta los parámetros anteriores, el carácter **=**.

Finalmente, ponga el permiso cuyos símbolos son: **r**, **w** o **x**.

Puede separar las modificaciones con comas y acumular varios permisos en un mismo comando.

```
$ ls -l
total 0
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic3
$ chmod g+w fic1
$ ls -l fic1
-rw-rw-r-- 1 seb users 0 mar 21 22:03 fic1
$ chmod u=rwx,g=x,o=rw fic2
$ ls -l fic2
-rwx--xrw- 1 seb users 0 mar 21 22:03 fic2
$ chmod o-r fic3
```

```
$ ls -l fic3
-rw-r----- 1 seb users 0 mar 21 22:03 fic3
```

Si quiere suprimir todos los permisos, no especifique nada después del signo =:

```
$chmod o=fic2
$ ls -l fic2
-rwx--x--- 1 seb users 0 mar 21 22:03 fic2
```

b. Sistema octal

La sintaxis es idéntica a la de los símbolos. A cada permiso le corresponde un valor octal, posicional y acumulable. Para codificar tres permisos rwx, hacen falta tres bits: cada uno tomaría el valor 0 o 1 según la presencia o ausencia del permiso. 23= 8, de ahí la notación octal.

- r vale 4.
- w vale 2.
- x vale 1.

La tabla siguiente servirá de ayuda:

Propietario			**Grupo**			**Resto de la gente**		
r	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**
400	200	100	40	20	10	4	2	1

Para obtener el permiso final, basta sumar los valores. Por ejemplo, si quiere rwxrw-rw-, entonces obtiene 400+200+100+40+10+4+1=755, y para rw-r--r-- 400+200+40+4=644.

```
$ chmod 755 fic1
$ chmod 644 fic2
$ ls -l fic1 fic2
-rwxr-xr-x 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
```

La modificación octal de los permisos no es sutil y no permite modificar un solo permiso. Es la totalidad de los permisos lo que se ha modificado de una sola vez.

3. Máscara de permisos

a. Restringir permisos de manera automática

En el momento de la creación de un archivo o de un directorio, se les asignan unos permisos automáticamente. Suele ser rw-r--r-- (644) para un archivo y rwxr-xr-x (755) para un directorio. Una máscara de permisos controla estos valores. Se puede modificar con el comando **umask**. El comando **umask** coge como parámetro un valor octal cuyo permiso individual se suprimirá de los permisos de acceso máximo del archivo o del directorio.

- Por defecto, se crean todos los archivos con los permisos 666 (rw-rw-rw-).
- Por defecto, se crean todos los directorios con los permisos 777 (rwxrwxrwx).
- Luego se aplica la máscara.
- La máscara es la misma para el conjunto de los archivos.
- Una máscara no modifica los permisos de los archivos existentes, sino solamente los de los archivos creados a partir de este momento.

Los permisos por defecto (máximo) de los archivos y de los directorios no son idénticos. Es lógico: como el permiso x permite entrar en un directorio, es normal que éste disponga de él por defecto. Este mismo permiso es inútil por defecto en los archivos: sólo una pequeña minoría de los archivos son scripts o binarios.

La máscara por defecto es 022, o sea `----w--w-`. Para obtener este valor, inserte **umask** sin parámetro.

```
$ umask
0022
```

b. Cálculo de máscara

Para un archivo

```
Predeterminado rw-rw-rw- (666)
Retirar        ----w--w- (022)
Resta          rw-r--r-- (644)
```

Para un directorio

```
Predeterminado rwxrwxrwx (777)
Retirar        ----w--w- (022)
Resta          rwxr-xr-x (755)
```

Observe que aplicar una máscara no es sustraer, sino suprimir permisos de los establecidos por defecto, permiso a permiso. Por ejemplo:

```
Predeterminado rw-rw-rw- (666)
Retirar        ----wxrwx (037)
Resta          rw-r----- (640)
```

Y no 629, lo que es imposible en sistema octal...

4. Cambiar de propietario y de grupo

Es posible cambiar el propietario y el grupo de un archivo gracias a los comandos **chown** (*change owner*) y **chgrp** (*change group*). El parámetro `-R` cambia la propiedad de manera recursiva.

```
chown usuario fic1 [Fic2...]
chgrp grupo fic1 [Fic2...]
```

Al especificar el nombre de usuario (o de grupo), el sistema comprueba primero su existencia. Usted puede especificar un UID o un GID. En este caso, el sistema no efectuará comprobación alguna.

Para los dos comandos, no se modifican los permisos anteriores ni la ubicación del archivo. Con un solo comando se puede modificar el propietario y el grupo a la vez.

```
chown usuario[:grupo] fic1 [fic2...]
chown usuario[.grupo] fic1 [fic2...]
```

Sólo root tiene permiso para cambiar el propietario de un archivo. Pero un usuario puede cambiar el grupo de un archivo si forma parte del nuevo grupo.

```
$ chgrp video fic1
$ ls -l fic1
-rwxr-xr-x 1 seb video 0 mar 21 22:03 fic1
```

5. Permisos de acceso extendidos

a. SUID y SGID

Es posible establecer **permisos de acceso especiales** para archivos ejecutables. Estos permisos de acceso extendidos aplicados a un comando permiten sustituir los permisos otrogados al usuario que lo inició por los permisos del propietario o del grupo a los que pertenece el comando.

El ejemplo más sencillo es el programa **passwd**, que permite cambiar la contraseña. Si se ejecutara el comando con los permisos de un usuario clásico, **passwd** no podría abrir y modificar los archivos **/etc/passwd** y **/etc/shadow**:

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1440 feb 24 10:35 /etc/passwd
```

Puede observar que este archivo pertenece a root, y que sólo root puede escribir en él. Un usuario normal no puede leer su contenido sin interactuar. El comando **passwd** no debería, por lo tanto, poder modificar los archivos. Vea los permisos del comando **passwd** (/bin/passwd o /usr/bin/passwd):

```
> ls -l /usr/bin/passwd
-rwsr-xr-x 1 root shadow 78208 sep 21 23:06 /usr/bin/passwd
```

Lleva asociado un nuevo permiso: **s** para los permisos del usuario root. Este nuevo atributo permite la ejecución del comando con permisos de acceso extendidos. Durante el tratamiento, se ejecuta el programa con los permisos del propietario del archivo o del grupo al que pertenece. En el caso de passwd, se inicia con los permisos de root y no del usuario que lo lanzó.

El permiso s sobre el usuario se llama **SUID-Bit** (*Set User ID Bit*), y sobre el grupo, **GUID-Bit** (*Set Group ID Bit*).

El comando **chmod** permite ubicar SUID-Bit y GUID-Bit.

```
chmod u+s comando
chmod g+s comando
```

Los valores octales son 4000 para SUID-Bit y 2000 para GUID-Bit.

```
chmod 4755 comando
chmod 2755 comando
```

Sólo el propietario o el administrador puede activar esta propiedad. Posicionar SUID-bit o SGID-Bit tiene sentido únicamente si se han establecido los permisos de ejecución previamente (atributo x en el propietario o el grupo). Si éstos no están presentes; se sustituye la **s** por una **S**.

b. Real / efectivo

En los datos de identificación del proceso, ha podido observar la presencia de **UID** y **GID reales y efectivos**. Cuando se inicia un comando con un SUID-Bit o un SGID-Bit posicionado, los permisos se modifican. El sistema conserva los UID y GID de origen del usuario que inició el comando (UID y GID reales) transmitidos por el padre, los números UID y GID efectivos son los del propietario o del grupo de pertenencia del programa.

P. ej.: pepito (UID=100, GID=100) envía passwd, que pertenece a root (UID=1, GID=1) con SUID-Bit

activado.

```
UID real: 100
GID real: 100
UID efectivo: 1
GID efectivo: 100
```

Si se posiciona sólo SGID-Bit:

```
UID real: 100
GID real: 100
UID efectivo: 100
GID efectivo: 1
```

Hay que subrayar que no se transmiten los SUID-Bit y SGID-bit a los hijos de un proceso. En este caso, se ejecutarán los hijos con los permisos del usuario que inició el comando básico, los UID reales.

c. Sticky bit

El **sticky bit** (*bit pegajoso*) permite asignar un criterio protector contra el borrado del contenido de un directorio. Imagine un directorio /tmp donde todos los usuarios tienen permiso para leer y escribir archivos.

```
$ ls -ld /tmp
drwxrwxrwx 6 root system 16384 Ago 14 13:22 tmp
```

En este directorio todo el mundo puede suprimir archivos, incluidos los que no le pertenecen (permiso w presente en todas partes y para todos). Si el usuario pepito crea un archivo, el usuario titi puede suprimirlo incluso aunque no le pertenezca.

El sticky bit aplicado a un directorio, aquí /tmp, impide esta operación. Sí, pepito aún puede visualizar y modificar el archivo, pero sólo su propietario (o el administrador) podrá suprimirlo.

```
$ chmod u+t /tmp
ls -ld /tmp
drwxrwxrwt 35 root root 77824 mar 21 22:30 /tmp
```

En octal, se utilizará el valor 1000 (`chmod 1777 /tmp`).

Aunque aplicado al usuario, el sticky bit, representado por una **t**, aparece en el grupo de permisos de "others".

d. Permisos y directorios

Si da el permiso s al grupo en un directorio, todos los archivos creados dentro de este directorio serán del mismo grupo que este directorio, sea cual sea el grupo de la persona que crea este archivo.

```
$ mkdir dir
$ chmod 770 dir
$ ls -ld dir
drwxrwx--- 2 seb users 4096 mar 21 22:36 dir
$ chgrp video dir
$ chmod g+s dir
$ ls -ld dir
drwxrws--- 2 seb video 4096 mar 21 22:37 dir
$ cd dir
```

```
$ touch pepito
$ ls -l pepito
-rw-r--r-- 1 seb video 0 mar 21 22:37 pepito
```

Proceso de inicio

1. La BIOS y UEFI

a. BIOS

La **BIOS** (*Basic Input Output System*) es un software de interfaz entre el hardware y el software a un nivel muy básico. Proporciona el conjunto de instrucciones básicas utilizadas por el sistema operativo. Provee el nivel de interfaz más bajo a los drivers y periféricos.

La BIOS está en la memoria **EEPROM** (*Electrical Erasable Programmable Read-Only Memory*) del ordenador. Cuando se enciende el ordenador, o tras un reseteado, se manda una señal llamada*powergood* al microprocesador, que activa la ejecución de la BIOS.

La BIOS efectúa un autochequeo del encendido (POST), luego busca los periféricos, en particular los utilizados para iniciar el sistema. A continuación, se almacena la información relativa al hardware de manera permanente en una pequeña memoria CMOS alimentada por una batería. Al final de proceso, se selecciona el periférico de inicio.

La BIOS lee y ejecuta el primer sector físico del soporte físico de inicio. Suele tratarse de los 512 primeros bytes del primer disco duro (el MBR) o de la partición activa (la PBR), como se describió en el capítulo Los discos y el sistema de archivos.

b. UEFI

El sucesor de BIOS se llama **UEFI** (*Unified Extensible Firmware Interface*, interfaz extensible de firmware unificada), sucesor de EFI. UEFI se pronuncia como "Unify" pero sin la n. Linux soporta UEFI pero el gestor de arranque debe ajustarse en consecuencia. La mayoría de los PC y placas base vendidos desde el inicio de los años 2010 disponen de UEFI por defecto.

UEFI es un componente software que realiza la interfaz entre el firmware del hardware y el sistema operativo. Garantiza independencia entre el sistema operativo y la plataforma hardware. Su función principal es arrancar el sistema operativo a través de un "boot manager". Al contrario que BIOS, que busca un programa de arranque en un MBR, UEFI cargará, mediante su microcódigo, todo lo necesario (controladores por ejemplo) para poder iniciar directamente la carga del SO, siempre que éste sea compatible con UEFI.

UEFI dispone de muchas ventajas y funcionalidades adicionales en comparación con BIOS (que ha quedado totalmente obsoleto para los procesadores, controladores y sistemas operativos modernos), tales como:

- una interfaz gráfica;
- soporte para discos de más de 2 TB empleando GPT;
- muchos servicios para gestionar variables (empleando la NVRAM), temporizador (RTC, almacenamiento del huso horario), consolas gráficas, arranque de SO, siendo algunos de estos servicios accesibles desde el SO;
- una selección de diferentes métodos de arranque: UEFI (para arrancar en GPT), CSM (*Compatibility Support Module*, para arrancar desde un MBR), red (PXE, TFTP, DHCP, IP, UDP), y arranque seguro.

Sobre este último punto se ha escrito mucho. El objetivo es impedir que UEFI arranque un sistema operativo o un gestor de arranque que no sea seguro, es decir, que no esté firmado correctamente con una clave verificada; un virus, un rootkit o un SO comprometido no podrán arrancar con un boot seguro.

El problema es que Microsoft anunció que sólo certificaría los equipos que dispusieran de Secure Boot y la clave privada de Microsoft para Windows 8. Por tanto, en estos equipos es imposible arrancar otro SO como Linux, hasta que Microsoft aclare su postura. Muchos fabricantes de Linux

han tenido que certificar su sistema de arranque, y por ende comprar una licencia a Microsoft para permitir que su SO arranque en estos equipos. Otros han decidido utilizar métodos alternativos, como el desarrollo de un mini bootloader firmado llamado Shim, un intermediario, compatible con Microsoft, que permite arrancar los SO firmados por sus fabricantes respectivos.

Felizmente, existen muchos UEFI que pueden emular el comportamiento de un BIOS y pueden continuar arrancando en un MBR clásico. Aun así, GRUB2 simula un arranque MBR en GPT. Además, la mayoría de los parámetros siguientes funcionan con UEFI.

Observe que UEFI es una norma respetada por la mayoría de los fabricantes, pero algunos lo modifican, lo que puede introducir incompatibilidades. Otros, por el contrario, son estrictos. El fabricante de placas base Gigabyte tiene una gran reputación por la calidad de su implementación UEFI, lo que permite por ejemplo arrancar Mac OS X.

c. Ajustes básicos

Cada BIOS es diferente según los fabricantes de tarjetas madre y los editores (AMIBios, Phoenix, Award, etc.). Sin embargo, muchos de los ajustes son idénticos o, en todo caso, se parecen mucho.

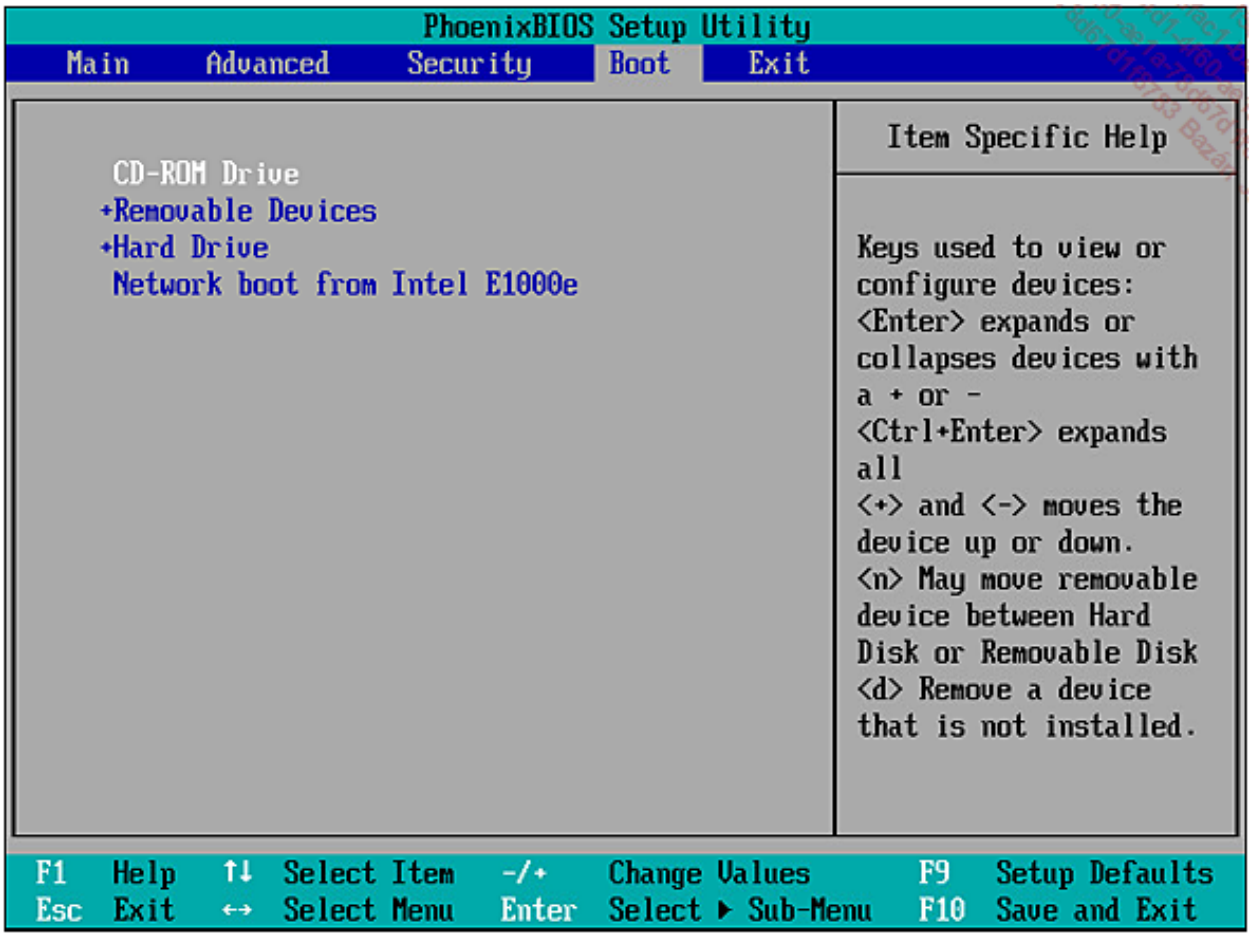
La detección de los discos duros y la elección del soporte de inicio se efectúan desde la BIOS. Linux soporta discos IDE, SATA y SCSI. No obstante, es posible que su juego de chips no reconozca el disco SATA. En este caso, la mayoría de las BIOS tienen como opción emular un IDE modificando el controlador de la SATA. Linux los reconocerá como tales. Es recomendable, sin embargo, probar una primera instalación con el soporte nativo de los discos SATA activado.

En principio, Linux gestiona correctamente el soporte de los chipsets SATA compatibles **AHCI**(*Advanded Host Controller Interface*), un estándar para las especificaciones públicas. Active esta opción en la BIOS siempre que pueda. En ocasiones aparece con este nombre y, en otras, en modo **nativo**. Si no funciona nada, intente el modo **combined**, y luego **legacy IDE**. Encontrará ayuda sobre el SATA en: http://ata.wiki.kernel.org/

Para ejecutar la instalación de Linux desde un soporte óptico (CD-Rom o un DVD-Rom) debe modificar el orden de ejecución de manera que arranque primero desde el lector de CD o DVD, lo que también es válido para los lectores Blu-ray.

Si su teclado es de tipo USB, o sin cable pero con un adaptador sin cable USB, debe activar el **USB legacy support** (a veces esta función se llama **USB DOS function o USB keyboard enable**). Permite activar en el momento del inicio el soporte de los teclados, pero también los soportes de almacenamiento (pendrives, discos duros, tarjeta de memoria). Esto no impide que el sistema se encargue del USB: una vez iniciado el SO, los drivers USB del núcleo y de los módulos se encargan del USB.

En principio, no tiene por qué tocar la configuración, los experimentos, con gaseosa: no modifique la configuración avanzada del chipset y de otros recursos cuya utilidad no entienda. Sin embargo, con el fin de ahorrar recursos, puede desactivar los puertos de la placa base que no utiliza: puerto paralelo, puerto serie, etc.



Pantalla de BIOS Phoenix para modificar el orden de arranque

El overclocking necesita un hardware especial: procesador, placa base, memoria y alimentación deben ser de alta calidad y el PC debe estar bien aireado. El overclocking es fuente de inestabilidad y cuelgues, tanto en Windows como en Linux. En particular, pone a prueba la memoria. Es la principal causa de inestabilidad. Incluso sin overclocking, resulta útil invertir en componentes de calidad.

2. El gestor de arranque

La BIOS activa el gestor de arranque inicial (*Initial Program Loader*, IPL) a partir de los primeros 512 bytes del soporte de arranque. En Linux, el gestor se divide en dos partes. El gestor inicial de los 512 bytes no contiene suficiente código para proponer menús y ejecutar el sistema operativo. Carga un segundo gestor basado en un archivo de configuración.

Este segundo gestor dispone de una interfaz para ejecutar un sistema operativo de entre una selección dada. Puede aprovechar este gestor para pasarle parámetros al núcleo Linux y al proceso init.

La BIOS sólo interviene en el inicio de la máquina, durante la utilización del gestor de arranque y en las primeras etapas de carga del núcleo. Luego, no sirve para nada más. El núcleo dispone de sus propias funciones de detección, aunque se apoye en la configuración de la BIOS. En efecto, ésta, bajo la plataforma Intel, se ejecuta en modo real, y Linux, en modo protegido.

Con GPT, el funcionamiento es ligeramente diferente si deseamos utilizar GRUB2, lo que se explicará en la parte correspondiente.

3. GRUB

a. Configuración

En muchas de las distribuciones Linux, como Red Hat 5 y 6 Entreprise, el gestor de arranque por defecto se llama **GRUB** (*Grand Unified Bootloader*). Tienen muchas posibilidades de configuración, en particular la protección con contraseña encriptada. Cuenta con un intérprete de comandos y con una interfaz gráfica. GRUB guarda las configuraciones en un archivo de texto y no es necesario volver a instalarlo tras cada modificación.

A continuación presentamos un ejemplo de configuración a partir del supuesto de que la primera partición del primer disco es /boot y que la segunda tiene una instalación de Windows.

```
timeout=10
default=0
title Red Hat
    root (hd0,0)
    kernel /vmlinuz-2.6.12-15 ro root=LABEL=/
    initrd /initrd-2.6.12-15.img
title Windows XP
    rootnoverify (hd0,1)
    chainloader +1
```

Esta tabla le muestra la sintaxis general de un archivo GRUB:

Parámetro GRUB	Significado
timeout	Número de segundos antes del arranque por defecto.
default n	Arranque por defecto (0=primer título, 1=segundo título, etc.).
gfxmenu	Ruta hacia un menú gráfico.
title xxxx	Principio de una sección, entrada del menú de GRUB.
root(hdx,y)	Se especificarán todos los accesos más abajo a partir de esta partición (ver el significado más adelante). Aquí, hd0,0 representa la primera partición del primer disco detectado por la BIOS. Es la partición /boot.
kernel	El nombre de la imagen del núcleo de Linux, seguido de sus parámetros. La / no indica la raíz del sistema de archivos, sino la de (hd0,0), por lo tanto /boot/vmlinuz...
initrd	Initial ramdisk. Imagen de disco en memoria que contiene la configuración y drivers iniciales y que el sistema sustituirá cuando cargue los discos definitivos.
rootnoverify	La raíz especificada. No debe montarse con GRUB (no soporta NTFS).
chainloader +1	Inicia el primer sector de la raíz especificado más arriba.

Damos el significado de los nombres de periféricos en GRUB.

- (fd0): primer lector de disquetes detectado por la BIOS (/dev/fd0 en Linux).
- (hd0,0): primera partición en el disco duro detectado por la BIOS, ya sea IDE o SCSI (/dev/hda1 o /dev/sda1 según los casos).
- (hd1,4): quinta partición en el segundo disco duro detectado por la BIOS (/dev/hdb5 o /dev/sda5).

b. Instalación

La configuración de **GRUB** reside en **`/etc/grub.conf`** o **`/boot/grub/menu.lst`** (el primero es

un vínculo al otro). El binario GRUB puede estar instalado en el **MBR** (*Master Boot Record*, los primeros 512 bytes de un disco) o un **PBR** (*Partition Boot Record*, los primeros 512 bytes de una partición).

Para instalar o desinstalar GRUB en caso de que MBR esté corrompido, por ejemplo en **/dev/sda**, utilice el comando **grub-install**:

```
# /sbin/grub-install /dev/sda
```

c. Arranque y edición

Al iniciarse GRUB aparece un menú. Puede ser gráfico o textual, según la configuración. Tiene que elegir una imagen de inicio entre las propuestas con las flechas de dirección. Pulsando la tecla [Entrar], ejecuta la imagen seleccionada.

Puede editar los menús directamente para modificar, por ejemplo, los parámetros pasados al núcleo de Linux o init. En este caso, seleccione una entrada de menú y pulse la tecla e (edit). Aquí se visualizan todas las líneas de la sección. Puede pulsar:

- e: para editar la línea (completarla);
- d: para suprimir la línea;
- o: para añadir una línea;
- b: para iniciar la imagen (booter).

Por ejemplo, para iniciar en modo emergencia (emergency):

- Vaya a la línea Linux o Red Hat y pulse **e**.
- Vaya a la línea kernel y pulse **e**.
- Al final de la línea añada 1 o Single y pulse [Entrar].
- Teclee **b**.

Puede acceder también a un intérprete de comandos pulsando [Esc]. ¡Cuidado! Sólo se reconocen los comandos GRUB.

4. GRUB2

a. GRUB2, el sustituto de GRUB

GRUB2 es el sucesor de GRUB y se ha reescrito desde cero. De su predecesor sólo conserva el nombre. Ya es el gestor de arranque por defecto de todas las distribuciones recientes de Ubuntu desde la versión 9.10 o Debian desde la versión 6. Está dotado de una interfaz gráfica (opcional), y es modular, compatible con varias arquitecturas (BIOS, EFI, Raid, etc.) y dispone de un modo de recuperación. Sin embargo, su configuración es un poco más difícil de comprender.

Algo importante que hay que recordar siempre es que la mayor parte de la configuración es automática. Los componentes de GRUB2 detectarán automáticamente la presencia de nuevos núcleos de Linux y de otros sistemas operativos como Windows, gracias al comando **os-prober**. Esta detección no se realiza en el arranque, sino en la ejecución de un comando de actualización.

Su instalación es idéntica a la de la primera versión de GRUB. Se puede migrar de GRUB a GRUB2 utilizando el comando **upgrade-from-grub-legacy**: el archivo menu.lst pasará a estar controlado y se aplicará la nueva configuración.

b. Configuración

Los archivos de GRUB2 siempre están en **/boot/grub** o **/boot/grub2**, según las distribuciones. Es

conveniente adaptar las rutas siguientes según sea el caso. Los nombres de módulos acaban con el sufijo .mod. Hay muchos, especialmente para la gestión del arranque en un gran número de sistemas de archivos o de tipos de particiones.

La configuración de las entradas de menú está en **/boot/grub/menu.cfg** o**/boot/grub2/grub.cfg**, según la distribución. Sin embargo, no debe modificar este archivo a mano, ya que debe generarse con el comando **update-grub** o **grub2-mkconfig**.

En la ejecución de **grub2-mkconfig**, se analizarán varios archivos para generar la configuración:

- **/bot/grub/device.map**, si existe, para la correspondencia entre los nombres de los discos GRUB y Linux.
- **/etc/default/grub**, que contiene los parámetros por defecto de GRUB.
- Todos los archivos albergados en **/etc/grub.d/**, por orden de lista, que son los scripts que permiten generar automáticamente los menús mostrados en el arranque.

Actualización de GRUB

A cada modificación de estos archivos (o cada vez que se añada un nuevo núcleo o script) debe seguirle la ejecución del comando **update-grub** o **grub2-mkconfig**:

```
# grub2-mkconfig -o /boot/grub/menu.cfg
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-2.6.38-8-generic
Found initrd image: /boot/initrd.img-2.6.38-8-generic
Found linux image: /boot/vmlinuz-2.6.35-28-generic
Found initrd image: /boot/initrd.img-2.6.35-28-generic
Found memtest86+ image: /memtest86+.bin
Found Windows 7 (loader) on /dev/sda1
done
```

device.map

El archivo device.map puede editarse manualmente o generarse automáticamente con el comando**grub-mkdevicemap**. Si el archivo device.map no está, la numeración de discos depende del orden de detección de éstos por la BIOS.

Si el comando no está, utilice **grub2-install -recheck**.

```
# grub-mkdevicemap
# cat /boot/grub/device.map
(fd0) /dev/fd0
(hd0) /dev/disk/by-id/ata-STM3500418AS_9VM2LMWK
(hd1) /dev/disk/by-id/ata-WDC_ED5000AACS-00G8B1_WD_WCAUK0742110
(hd2) /dev/disk/by-id/ata-WDC_WD10EADS-00L5B1_WD-WCAU4A320407
```

/etc/default/grub

El archivo **/etc/default/grub** contiene variables que definen las opciones de GRUB2 y opciones por defecto de los núcleos Linux.

A continuación se comentan algunas entradas:

Variable GRUB	Significado
GRUB_DEFAULT	Entrada del menú seleccionada por defecto (primera=0).
GRUB_HIDDEN_TIMEOUT	Considerando una duración en segundos, si esta variable está presente, el menú se

	oculta, lo que permite arrancar en el sistema predeterminado si sólo hay uno. Durante este intervalo de tiempo si el usuario pulsa [Esc] o [Shift] aparecerá el menú por pantalla.
GRUB_TIMEOUT	Duración de la visualización del menú en segundos antes de que arranque la entrada por defecto.
GRUB_HIDDEN_TIMEOUT_QUIET	Define si se visualiza o no el contador de timeout si GRUB_HIDDEN_TIMEOUT está definido: true o false.
GRUB_DISTRIBUTOR	Línea de comandos que define la generación automática del texto en la entrada del menú.
GRUB_CMDLINE_LINUX_DEFAULT	Parámetros pasados al núcleo Linux, únicamente para las entradas "normales", no las de recuperación.
GRUB_CMDLINE_LINUX	Parámetros por defecto para todos los núcleos LINUX.
GRUB_GFXMODE	Resolución utilizada por GRUB2 en modo gráfico, que puede modificarse por un valor devuelto por el comando **vbeinfo**.
GRUB_TERMINAL	Sin comentar y con el valor **console**, se activa el modo texto.

A continuación un ejemplo:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash nomodeset"
GRUB_CMDLINE_LINUX=""
GRUB_GFXMODE=640x480
```

Construcción de los menús

Los menús se construyen con los scripts albergados en **/etc/grub.d/**:

```
# ls -l /etc/grub.d/*
-rwxr-xr-x 1 root root 6658 2011-05-02 21:04 /etc/grub.d/00_header
-rwxr-xr-x 1 root root 5522 2011-04-21 15:19 /etc/grub.d/05_debian_theme
-rwxr-xr-x 1 root root 6291 2011-05-02 21:43 /etc/grub.d/10_linux
-rwxr-xr-x 1 root root 5233 2011-04-21 15:31 /etc/grub.d/20_linux_xen
-rwxr-xr-x 1 root root 1588 2010-09-24 20:16 /etc/grub.d/20_memtest86+
-rwxr-xr-x 1 root root 7119 2011-04-21 15:31 /etc/grub.d/30_os-prober
-rwxr-xr-x 1 root root  214 2010-10-06 14:22 /etc/grub.d/40_custom
-rwxr-xr-x 1 root root   95 2010-10-06 14:22 /etc/grub.d/41_custom
-rw-r--r-- 1 root root  483 2010-10-06 14:22 /etc/grub.d/README
```

Estos archivos se ejecutarán en el orden de listado (orden alfanumérico de forma predeterminada) de ls. Por este motivo empiezan con un valor numérico. Los dos scripts interesantes son **10_linux**y **30_os-prober**.

Si examina el primero, encontrará como elemento particular los bucles que buscan los núcleos Linux en /boot y en la raíz, así como los ramdisk iniciales (ver más adelante), para detectar los distintos núcleos Linux y crear automáticamente las entradas asociadas añadiendo los parámetros por

defecto.

En el segundo, verá que se ejecuta el comando **os-prober**, que busca el resto de sistemas operativos en el resto de discos y particiones, como Windows, Hurd, BDS, Mac OS X o incluso Linux. Lo puede comprobar usted mismo:

```
# os-prober
/dev/sda1:Windows 7 (loader):Windows:chain
```

Puede añadir sus propios scripts partiendo de alguna plantilla de las que vienen con el paquete (**40_custom**, por ejemplo). Lo que devuelve su script se añade automáticamente a grub.cfg, del cual hay que respetar su sintaxis.

c. Arranque y edición

El uso de GRUB2 en el arranque es muy parecido al de GRUB, pero con algunas sutilezas. Para editar alguna entrada use siempre la tecla **e** y modifique las distintas líneas como anteriormente. El editor se comporta como un miniEmacs. Se tendrá que usar [Ctrl] **x** para arrancar con sus modificaciones.

d. Caso de GPT y UEFI

Si su UEFI ofrece un modo de boot Legacy (emulación de BIOS), puede utilizar GRUB2 como gestor de arranque. Deberá crear una partición especial llamada **BIOS Boot partition** con un tamaño mínimo recomendado de 1 MB, identificado con el marcador (flag) **bios_grub** y no formateada. Con gdisk, utilice el código **ef02** como identificador de partición, el marcador se colocará automáticamente, lo que se puede comprobar con parted.

```
# parted /dev/sdb print
...
Number  Start      End         Size      File system    Name                  Flags
 1      1049kB     2097kB      1049kB                   BIOS boot partition  bios_grub
```

GRUB2 es totalmente compatible con el boot UEFI. Si desea emplear este modo, deberá crear una partición EFI de tipo **ef00**, montada en **/bott/efi**, de un tamaño de **200 MB**, y con un marcador**boot** y formateada en fat32. gdisk crea automáticamente el marcador, la creación del sistema de archivos la realiza usted.

```
# parted /dev/sdb print
...
Number  Start      End         Size      File system      Name               Flags
1       1049kB     211MB       210MB      fat32            EFI System        boot, esp
```

Este es el comando de creación del sistema de archivos FAT32:

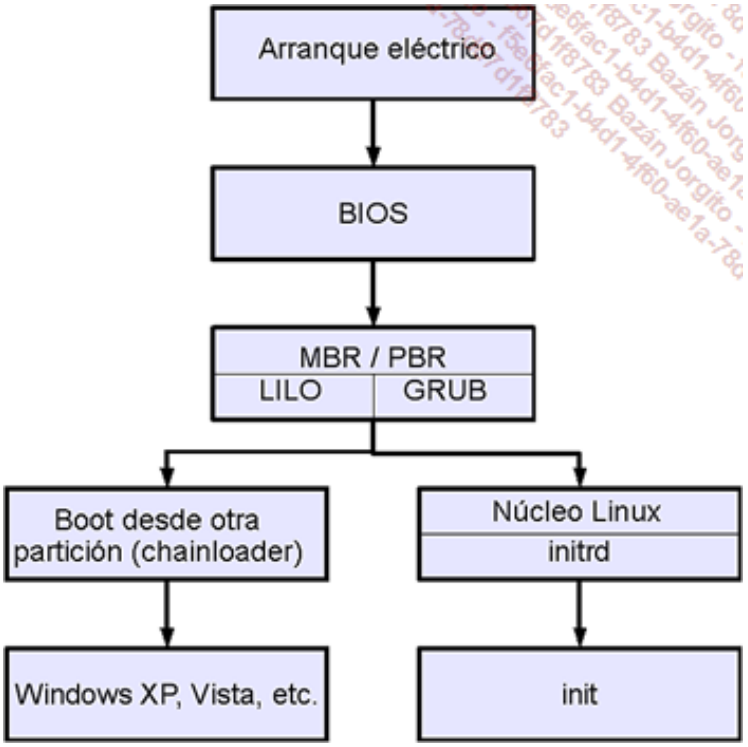
```
# mkfs.vfat -F 32 /dev/sdb1
```

5. Inicialización del núcleo

Durante la carga del núcleo, el monitor le mostrará una tonelada de información. El sistema no la registra en esta etapa, pero en la siguiente, la etapa del init, Linux comienza a escribir registros en el archivo **/var/log/dmesg**.

- Se detecta e inicializa el hardware.
- initrd está cargado, los módulos presentes se han cargado si es preciso.
- El núcleo monta el sistema de archivos raíz en modo de sólo lectura.

- Crea la primera consola.
- Se ha iniciado el primer proceso (en general init).



Esquema de la secuencia de arranque

Existen muchos gestores de arranque, como **LILO** (*Linux Loader*). Sin embargo, GRUB y GRUB2 los han sustituido (salvo en raras excepciones) casi por completo como consecuencia de sus numerosas limitaciones.

init System V

1. Funciones

El programa init es el primer proceso que se inicia y el último que se para dentro del sistema, y su misión es ejecutar las demás tareas. La función de init consiste en iniciar y parar todos los servicios. init ejecutará las diferentes tareas iniciales necesarias para el buen funcionamiento de Linux mediante la ejecución de varios comandos y scripts.

Una vez iniciado el sistema y ejecutados los servicios, init sigue activo para gestionar los cambios de estado de los procesos que controla y de los niveles de ejecución.

El programa init puede cambiar de una distribución a otra. En la mayoría de las distribuciones profesionales o mayores (Mandriva, Red Hat/Fedora, openSUSE, Debian, etc.), el principio sigue siendo más o menos el mismo: init de tipo System V (basado en la noción de niveles de ejecución). La distribución Ubuntu utiliza Upstart, que gestiona las prioridades, los eventos y dependencias entre los servicios, pero respeta el mismo principio. La distribución Slackware utiliza otro mecanismo inspirado en BSD.

Finalmente, la distribución Fedora utiliza desde la versión 15 systemd otro reemplazo de init, de diseño moderno y disponible únicamente para Linux. Hoy en día, systemd suele sustituir a init.

Sin embargo, la versión System V de init aún goza de días de gloria: no sólo la siguen usando la mayoría de distribuciones de Linux para empresas, sino que además su principio de diseño y su configuración son casi idénticos a las versiones Unix propietarias.

El proceso **init** es el padre de todos los procesos. Siempre tiene el PID 1. Su configuración está en el archivo **`/etc/inittab`**. Si este archivo está corrompido o es inutilizable, habrá que arrancar el sistema en modo single (S, s, 1 Single) y arreglarlo, o en el peor de los casos arrancar desde un soporte externo o un disco de emergencia. Es un archivo central del sistema operativo.

2. Niveles de ejecución

Un nivel de ejecución, o runlevel, corresponde al estado en el cual está Unix/Linux. Init controla dicho estado. Cada estado dispone de su propia configuración (o por inittab, o por scripts llamados initscripts). Por ejemplo, se puede utilizar un nivel de ejecución para arrancar Unix en modo monousuario, en multiusuarios, con o sin red, con o sin modo gráfico. El administrador puede personalizar todos los niveles. Por convención, en las distribuciones Red Hat/Fedora, Mandriva, openSUSE y asociadas, se suelen definir estos niveles de la siguiente manera:

Nivel	Efecto
0	Halt: detiene el sistema operativo, apaga la máquina.
1	Modo monousuario utilizado para el mantenimiento, modo consola.
2	Multiusuario, sin red, consola.
3	Multiusuario, con red, consola.
4	Igual que el 3, a conveniencia del administrador.
5	Multiusuario, con red, con entorno gráfico X Window.
6	Reboot: reinicio de la máquina.
S,s	Single user mode, el modo más bajo en caso de problema.

Los niveles 7 a 9 son perfectamente válidos, pero no se utilizan por defecto. El nivel de ejecución se sitúa por defecto en **`/etc/inittab`** en la línea initdefault.

```
id:5:initdefault:
```

Sustituya 5 por el nivel deseado en el momento del arranque.

Un reto en ciertos ejercicios eliminatorios de certificaciones consiste en crear una situación de avería en la cual el valor por defecto está en 0 o 6. En el primer caso, la máquina se apaga en cuanto se ejecuta init. En el otro, arranca en bucle...

La distribución Debian (y las distribuciones que derivan de ella) considera también los niveles 2 a 5 como multiusuario, pero no establece diferencias entre estos niveles. Por defecto, se arranca en el nivel 2, donde se inicia todo, incluso en su caso la interfaz gráfica.

Como se puede modificar y volver a configurar completamente cada nivel, es posible volver a definirlo todo, y por lo tanto, hacer que una Debian se inicie como una Red Hat, y viceversa. Por motivos de conformidad y asistencia, tenga en cuenta que es importante seguir el "estándar" de la distribución que utiliza.

3. /etc/inittab

Se define el comportamiento del proceso init y de los runlevels en el archivo **`/etc/inittab`**. La sintaxis de una línea es la siguiente:

```
Id:[niveles]:acción:comando
```

Campo	Descripción
Id	Identificador de línea sobre cuatro caracteres (en Linux con getty/mingetty: número de terminal).
Niveles	Indica si se debe tener en cuenta el comando para el nivel requerido. Corresponde a la lista de los niveles sin separador.
Acción	Tipo de acción que efectuar según las circunstancias para esta línea.
Comando	El comando que ejecutar con sus parámetros y redirecciones.

La acción es muy importante, ya que define las actividades de init durante el arranque y cambio de nivel.

Las principales se presentan a continuación:

Acción	Significado
initdefault	Define el nivel por defecto durante el boot y el inicio de init.
sysinit	Se ejecuta una única vez durante el arranque del sistema.
boot	Igual, pero después de sysinit.
bootwait	Igual, pero init espera el final de la ejecución del comando antes de seguir leyendo el archivo inittab.
off	Se ignora la línea.
once	Se ejecuta el comando a cada cambio de nivel para los niveles especificados.
wait	Igual, pero init espera que finalice la ejecución antes de proseguir.
respawn	El comando se ejecuta para los niveles correspondientes. Si el proceso termina, vuelve a arrancarse de nuevo automáticamente. Es lo que ocurre con los terminales si un usuario se desconecta.
powerwait	Inicia el comando si el servidor es alimentado mediante una fuente de

	emergencia (UPS).
powerfail	Igual, pero sin esperar a que finalice la ejecución del comando.
powerokwait	Se ejecuta el comando cuando se restablece la corriente.
powerfailnow	Comando de último recurso cuando la alimentación de emergencia está casi vacía.
ctrlaltdel	Init recibe una señal SIGINT procedente de una secuencia [Alt][Ctrl][Supr].

Damos un ejemplo procedente de una instalación openSUSE 10.3:

```
# Nivel de ejecución a 5 (multiusuario gráfico)
id:5:initdefault:

# Primer script ejecuta el arranque
si::bootwait:/etc/init.d/boot

# Gestión de los servicios por nivel de ejecución
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

# Caso del modo single, consola de emergencia para root
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin

# Acción en Alt+Ctrl+Del
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# Qué hacer en caso de corte de la corriente
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# Arranque de las consolas virtuales Alt+Fx
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

4. Cambio de nivel

También puede cambiar de nivel después de arrancar la máquina con el comando **/sbin/init** o**/sbin/telinit**, al ser este último un simple vínculo simbólico a init. El comando siguiente pasa al nivel 5.

```
# telinit 5
```

Se pueden especificar los valores `q`, `u` o `-t`:

- `Q` o `q`: init vuelve a leer el archivo **/etc/inittab**, si se ha modificado, y corrige sus tablas internas.
- `U` o `u`: init vuelve a iniciarse sin leer inittab y sin cambiar de nivel. Si se han añadido o suprimido servicios del nivel en curso, init tiene en cuenta la modificación.

- `-t`: cuando init termina la parada de los servicios (o más bien cuando lo hace el script rc; ver más adelante), init manda la señal SIGTERM a todos los procesos restantes, les pide que se terminen correctamente, espera el número de segundos especificado (5 por defecto), luego manda SIGKILL.

El nivel de ejecución es visible con el comando **/sbin/runlevel**. El primer valor devuelto corresponde al nivel que precede el nivel actual. Una N significa que no hay un nivel anterior. El segundo valor es el nivel actual.

```
# runlevel
N 5
```

5. Configuración del sistema básico

Sea cual sea el nivel de ejecución especificado por defecto, init inicia siempre el comando asociado a las acciones sysinit, bootwait o boot de **/etc/inittab** en el momento de arrancar el sistema. La acción sysinit es la primera.

- En Red Hat: `si::sysinit:`**/etc/rc.d/rc.sysinit**
- En openSUSE: `si::bootwait:`**/etc/init.d/boot**
- En Debian: `si::sysinit:`**/etc/init.d/rcS**

En Red Hat, es un script único monolítico el que se encarga de toda la configuración básica. En Debian, el script llama a todos los scripts del nivel S (single). En openSUSE, el script instala lo estrictamente necesario y luego ejecuta el contenido de **/etc/rc.d/boot.d**, que establece el resto de la configuración básica.

En todos los casos, las tareas siguientes se ejecutan aproximadamente en este orden:

- Configuración de los parámetros del núcleo presentes en **/etc/sysctl.conf** (p. ej.: IP Forwarding).
- Instalación de los archivos periféricos (/dev mediante udev, por ejemplo).
- Configuración del reloj del sistema.
- Carga de las tablas de caracteres del teclado.
- Activación de las particiones de intercambio SWAP.
- Definición del nombre de anfitrión.
- Control y montaje del sistema de archivos raíz (en lectura-escritura esta vez).
- Añadido de los periféricos RAID, LVM o ambos. Esto ya se puede instalar durante la carga de inittab.
- Activación de las cuotas de disco.
- Control y montaje de los demás sistemas de archivos.
- Limpieza de los bloqueos (stale locks) y de los archivos PID en caso de parada brusca.

En algunas distribuciones es posible interactual con init. Al principio del boot, después del arranque de init, le pueden pedir que pulse la letra i y que conteste sí o no a las diferentes acciones.

6. Nivel de ejecución

El script **/etc/init.d/rc** coge como parámetro el nivel de ejecución por defecto según la línea**initdefault** de **/etc/inittab** o el parámetro especificado durante la llamada manual de los comandos **init** o **telinit**. El script rc inicializa el nivel de ejecución deseado y es responsable del inicio y de la parada de los servicios asociados cuando el nivel de ejecución cambia.

```
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

Se analizan los servicios en cada nivel de ejecución. Durante el paso de un nivel a otro, y sea cual sea el orden (del 2 al 5, del 5 al 3, etc.) el script rc compara entre el antiguo y el nuevo nivel los servicios que se deben detener o iniciar. Si dos niveles tienen el mismo servicio, lo mantienen. Si se debe iniciar un nuevo servicio en el nuevo nivel, lo hace. Si se debe parar un servicio porque no está en el nuevo nivel, lo para.

Este funcionamiento, estándar a todas las distribuciones Linux de tipo System V, no es común en todos los Unix. HP-UX (un Unix de HP) considera que debe haber un progreso constante en los niveles, pasando sucesivamente del 1 al 3 (1 luego 2 luego 3) y cargando de manera sucesiva los servicios. Parado, baja hasta el nivel 0 y termina los servicios sucesivamente. La diferencia es importante: no compara los niveles y no efectúa parada/reinicio entre cada nivel...

7. Gestión de los niveles y de los servicios

a. Servicios en init.d

El nivel de ejecución define los servicios que se deben iniciar para este nivel. Le corresponde al script rc cargar los servicios. Se controlan los servicios (inicio, parada, reinicio, estatus, etc.) mediante scripts presentes en el directorio **/etc/init.d**.

```
# cd /etc/init.d
# ls -l
-rwxr-xr-x  1 root root  1128 ago  9  2004 acpid
-rwxr-xr-x  1 root root   834 sep 28  2004 anacron
-rwxr-xr-x  1 root root  1429 jun 22  2004 apmd
-rwxr-xr-x  1 root root  1176 abr 14  2006 atd
-rwxr-xr-x  1 root root  2781 mar  5  2007 auditd
-rwxr-xr-x  1 root root 17058 sep  5  2007 autofs
-rwxr-xr-x  1 root root  1368 feb  2  2007 bluetooth
-rwxr-xr-x  1 root root  1355 may  2  2006 cpuspeed
-rwxr-xr-x  1 root root  1904 jul 16  2007 crond
-rwxr-xr-x  1 root root  2312 oct 30 13:46 cups
...
```

Para cada nivel de ejecución n, existe un directorio rcn.d que contiene vínculos simbólicos (atajos) hacia los servicios presentes en **/etc/init.d** que se quieren iniciar o parar. Este directorio está en diferentes lugares según la distribución:

- Red Hat: **/etc/rc.d/rcn.d** con vínculos **en/etc/rcn.d**
- openSUSE: **/etc/init.d/rcn.d**, sabiendo que **/etc/rc**.d apunta a **/etc/init.d**
- Debian: **/etc/rcn.d**

El prefijo del nombre de cada vínculo define su orden de ejecución o de parada. El nombre está con la forma siguiente:

```
[SK]nnservicio
```

- **S**: start.

- **K**: kill (stop).
- **nn**: orden numérico de ejecución o parada (00=primero, 99=último).
- **servicio**: nombre del servicio.

Por ejemplo, el vínculo S10network indica que se iniciará el servicio network, responsable de la instalación de la red, en orden 10, después de los S01, S05, etc., pero antes de los S11, S15, S20, etc.

```
# ls -1 S*
S00microcode_ctl
S01sysstat
S02lvm2-monitor
S05kudzu
S06cpuspeed
S08iptables
S09isdn
S09pcmcia
S10network
S12syslog
S13irqbalance
S13portmap
S14nfslock
S15mdmonitor
S18rpcidmapd
...
```

Cuando se ejecuta rc, primero lista todos los vínculos que comienzan por K* usando un bucle for. Luego hace lo mismo para S*, y esta vez inicia los servicios. Presentamos una parte del archivo rc para entender mejor la secuencia de inicio:

```
# prueba de la existencia de /etc/rcn.d
if [ -d /etc/rc${level}.d ]
then
    # Lista todos los scripts que empiezan por S en este directorio
    for i in /etc/rc${level].d/S*
    do
        # el script existe y no está vacío: se ejecuta
        if [ -s ${i} ]
             then
                 sh ${i} start
        fi
    done
fi
```

b. Control manual de los servicios

Mediante scripts

Se pueden iniciar los servicios en todos los casos individualmente o con la ayuda de herramientas según la distribución. El primer método es el único por defecto en Debian.

Cada servicio presente en /etc/init.d acepta al menos dos parámetros:

- **start**: el servicio se inicia.
- **stop**: el servicio se para.

Si desea iniciar o parar el servicio sshd (servidor ssh) manualmente:

```
# /etc/init.d/sshd start
Starting SSH daemon                                        done
```

```
# /etc/init.d/sshd stop
Shutting down SSH daemon                                    done
```

Algunos servicios pueden aceptar otros parámetros:

```
# /etc/init.d/sshd
Usage: /etc/init.d/sshd {start|stop|status|try-
restart|restart|force-reload|reload|probe}
```

- **status**: facilita el estado del servicio (iniciado o no). Según los servicios, se puede facilitar información adicional.
- **probe**: indica si es necesario cargar la configuración; si, por ejemplo, se han modificado archivos de configuración.
- **reload / forcereload**: indica al servicio que vuelva a leer su configuración (mediante una señal 1).
- **restart**: espera y vuelve a iniciar el servicio, sea cual sea el final de la parada.
- **try-restart**: para y vuelve a iniciar el servicio solamente en caso de parada.

La distribución openSUSE crea vínculos simbólicos **rc<servicio>** que permiten insertar **rcsshd**, por ejemplo para el control manual de los servicios.

Mediante el comando service

El comando **service** está disponible en Red Hat y openSUSE. Permite simplemente prescindir de la ruta hacia el script de inicio del servicio y utilizar simplemente su nombre:

```
# service sshd stop
Shutting down SSH daemon                                    done
# service sshd start
Starting SSH daemon                                         done
```

Para controlar la configuración de los servicios de System V lanzados por init, no se aconseja hacerlo todo a mano, sino más bien utilizar las herramientas del sistema correspondiente cuando existan, tanto en modo texto o como gráfico.

c. Modificación de los niveles de ejecución

Red Hat y openSUSE

En Red Hat/Fedora y openSUSE, el comando **chkconfig** permite añadir, suprimir, activar o desactivar scripts, por nivel de ejecución. Este comando es muy práctico para configurar los servicios porque sabe gestionar tanto los servicios System V como los servicios xinetd.

```
chkconfig [opción] [servicio]
```

Aunque la sintaxis sea idéntica en las dos distribuciones, chkconfig no funciona de la misma manera. En Red Hat, se inserta una línea especial al principio del script que indica a chkconfig los parámetros por defecto (runlevels, posiciones de inicio y parada). En openSUSE, chkconfig es un frontend para el comando **insserv**. Este último aprovecha también el encabezamiento de los scripts, pero de manera más compleja (gestiona la ordenación y el paralelismo, por ejemplo).

Presentamos las primeras líneas del script de inicio de servicio sshd en Red Hat. El script se inicia y se para en los niveles 2, 3, 4 y 5. Se ejecuta en posición 55 (S55sshd) y se para en posición 25

(K25sshd).

```
# chkconfig: 2345 55 25
# description: OpenSSH server daemon
```

Veamos lo mismo en openSUSE. chkconfig e insserv gestionan ellos mismos el orden de inicio y parada gracias a los campos Required-Start y Required-Stop. Se deben iniciar los servicios de red y remote_fs antes de sshd. El servicio se inicia en los niveles 3 y 5 y se detiene en los niveles 0 (parada), 1 (single user), 2 (sin red) y 6 (reboot).

```
### BEGIN INIT INFO
# Provides: sshd
# Required-Start: $network $remote_fs
# Required-Stop: $network $remote_fs
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Description: Start the sshd daemon
### END INIT INFO
```

Damos la lista de las opciones de chkconfig:

- **--list**: lista del conjunto de la configuración.
- **--list** servicio: la configuración de un servicio dado.
- **--add** servicio: añade el servicio indicado en la configuración System V.
- **--del** servicio: suprime el servicio de la configuración System V.
- **--level xxx** servicio **on/off**: activa o desactiva el servicio para los niveles de ejecución indicados.

```
# chkconfig --list
rwhod    0:parada 1:parada   2:parada   3:parada   4:parada   5:parada   6:parada
atd      0:parada 1:parada   2:parada   3:funciona 4:funciona 5:funciona 6:parada
snmpd    0:parada 1:parada   2:funciona 3:funciona 4:funciona 5:funciona 6:parada
ntpd     0:parada 1:parada   2:funciona 3:funciona 4:funciona 5:funciona 6:parada
keytable 0:parada 1:funciona 2:funciona 3:funciona 4:funciona 5:funciona 6:parada
syslog   0:parada 1:parada   2:funciona 3:funciona 4:funciona 5:funciona 6:parada
...

# chkconfig --list smb
smb      0:parada 1:parada   2:parada   3:parada   4:parada   5:parada   6:parada

# chkconfig --level 35 smb on
# chkconfig --list smb
smb      0:parada 1:parada   2:parada   3:funciona 4:parada   5:funciona 6:parada

# /sbin/chkconfig --add httpd
```

chkconfig no inicia ningún servicio. Sólo configura los niveles de ejecución. Para iniciar un servicio, se utilizará el script asociado o el comando servicio.

En Debian y Ubuntu

El comando **update-rc.d** crea los vínculos necesarios en los distintos directorios rcn.d, un poco como chkconfig, pero de manera más "bruta": los scripts no tienen la información particular en su encabezamiento y el comando sólo funciona a nivel del sistema de archivos, por lo que instala los diferentes vínculos según sus indicaciones.

Veamos dos ejemplos de sintaxis. El primero inscribe un servicio con parámetros por defecto. En

este caso, se configura el servicio para ejecutarse del nivel 2 a 5 y pararse en los niveles 0, 1 y 6. La posición de parada/relanzamiento está en 20.

```
# update-rc.d ssh defaults
```

En el segundo ejemplo, se inserta el servicio con opciones completas. Al iniciarse, el servicio está en posición 10 en los niveles 3, 4 y 5. Parado, el servicio está en la posición 5 en los niveles 0, 1 y 6. No olvide los puntos.

```
# update-rc.d ssh start 10 3 4 5 . stop 05 0 1 6 .
```

El parámetro remove suprime los vínculos de los diversos directorios. Sin embargo, el propio script **/etc/init.d/xxx** asociado ya no debe existir. En caso contrario, utilice el parámetro `-f` para forzar la supresión de los vínculos (el propio script sigue en su sitio).

```
# update-rc.d -f ssh remove
```

8. Consolas virtuales

Las consolas virtuales permiten obtener terminales virtuales en una máquina. Se definen en **/etc/inittab**. Están disponibles mediante los periféricos **/dev/ttyn**, donde n es el número de consola.

No se suele instalar ni iniciar la capa gráfica en los servidores de empresa. Se accede al servidor o bien directamente (o por kvm), o bien por la red (ssh por ejemplo) o mediante soluciones integradas (puertos de administración del servidor, como ILO en las máquinas HP).

- Pase de una consola a la otra con la secuencia de teclas [Alt][Fn] (p. ej.: [Alt][F2]) desde la consola o [Ctrl][Alt][Fn] desde X Window.
- Utilice las teclas [Alt][Flecha derecha] y [Alt][Flecha izquierda] para pasar a la consola siguiente o a la anterior.
- /dev/ttyn representa la consola virtual n.
- /dev/tty0 representa la consola actual.
- Como hay 12 teclas de función, puede haber por defecto 12 terminales virtuales.
- Sin embargo, se activan "solamente" 6 por defecto.
- Se inicia X Window por defecto en la primera consola disponible, en general la 7.

Se inician las consolas con inittab y los procesos **getty** o **mingetty**. Son las únicas entradas de inittab donde la etiqueta tiene su importancia: corresponde al número de la consola. Observe la utilización de respawn. Como la consola establece la petición de login, luego el shell se sustituye y, cuando se termina, se inicia un proceso mingetty automáticamente para aceptar una nueva conexión.

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

9. Los logins

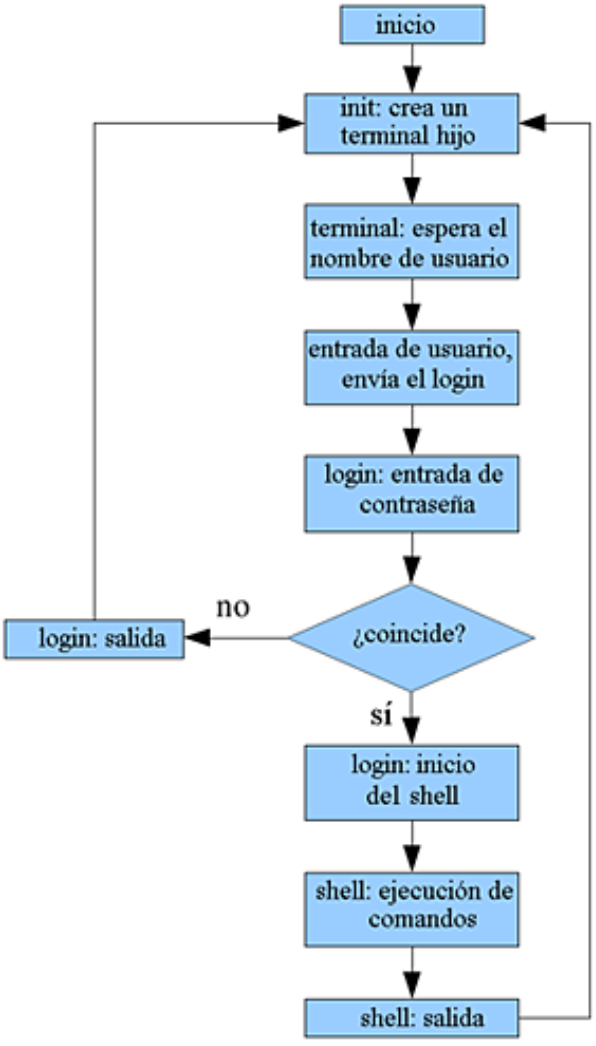
Una vez que init ha iniciado los terminales (getty), un prompt espera la inserción del nombre (el login) del usuario. Antes de este prompt, se visualiza el contenido del archivo **/etc/issue**. Con el nombre validado, el terminal ejecuta el comando **login**, que requiere la inserción de la contraseña. Si la contraseña es correcta (verificación en **/etc/passwd** y **/etc/shadow** o utilización de los módulos PAM), entonces login muestra el contenido de **/etc/motd** y ejecuta un shell (siempre definido

en **/etc/passwd**).

Observe que getty y login no efectúan un fork: los procesos iniciados no son hijos, pero solapan el proceso actual (API exec). No hay relaciones hijos-padres entre los procesos (getty->login->shell) sino que cada uno sustituye al anterior, y el proceso guarda el mismo PID. De esta manera init sabe cuándo se termina una conexión.

Una vez terminada la sesión (fin del shell), init vuelve a iniciar un terminal para una nueva conexión (comando respawn).

getty va a permitir un buen funcionamiento del terminal del usuario: para ello se va a adaptar a sus diferentes parámetros (VT100, VT220, XTERM, CONSOLE...). Además, getty puede escuchar perfectamente un puerto serie y soportar una conexión con módem (e iniciar luego una sesión ppp, por ejemplo).



Secuencia de inicio de sesión empleando login

10. Parada

Varios métodos permiten parar correctamente una máquina en Linux. Antes de nada y a título orientativo, init gestiona las paradas con los niveles 0 y 6. En la práctica, los dos son casi idénticos salvo para la última acción.

- Runlevel 0: el ordenador se apaga automáticamente.
- Runlevel 6: el ordenador se apaga y se reinicia de nuevo.

De esta manera, el comando siguiente apaga el ordenador:

```
# init 0
```

Este comando vuelve a iniciarlo:

```
# init 6
```

Sin embargo, el comando más correcto, más limpio y más seguro para parar el sistema es **shutdown**. Shutdown llama a init, pero acepta parámetros adicionales. Su sintaxis básica es: `shutdown <param> <plazo> <mensaje>`.

Los parámetros son:

Parámetro	Acción
`-k`	No apaga el sistema, sino que manda el mensaje de apagado a todo el mundo.
`-r`	Reinicio.
`-h`	(halt) Parada.
`-f`	Impide la ejecución de fsck en el boot.
`-F`	Fuerza la ejecución de fsck en el boot.
`-c`	Cancela el shutdown sin demora, pero puede enviar un mensaje.

Se puede especificar el plazo de diferentes maneras:

- **hh:mm**: una hora precisa.
- **+m**: en m minutos.
- **now**: un alias para +0, es decir, ahora mismo.

El ejemplo siguiente programa un reinicio para dentro de 10 minutos con un mensaje de aviso.

```
# shutdown -r +10 "Reinicio para mantenimiento en 10 minutos"

Broadcast message from root (pts/2) (Fri Apr 4 15:00:34 2008):

Reinicio para mantenimiento en 10 minutos
The system is going DOWN for reboot in 10 minutes!
```

El ejemplo siguiente cancela el reinicio.

```
# shutdown -c "Mantenimiento cancelado"

Shutdown cancelled.

Broadcast message from root (pts/2) (Fri Apr  4 15:02:21 2008):

Mantenimiento cancelado
```

Se llama a los comandos **reinicio** y **halt** al final de init 6 y 0, respectivamente. Si se llaman a un nivel distinto de 6 o 0, equivale a una llamada de shutdown:

- **halt**: shutdown -h
- **reboot**: shutdown -r

systemd

1. Fundamentos

El sustituto de init **systemd** es fruto de una reflexión profunda sobre la ejecución de servicios, combinando todos los sistemas operativos. Su autor ha comparado en particular init System V, upstart y el sistema de arranque de Mac OS X y ha sacado conclusiones interesantes sobre las ventajas y limitaciones de cada uno de ellos. Se puede acceder a sus conclusiones en la siguiente página web: http://0pointer.de/blog/projects/why.html

Son bastante constructivas.

En su defensa, su autor hace algunas constataciones interesantes sobre la duración del arranque de un sistema y el número de procesos que arrancan antes de llegar a la pantalla de login. Especialmente, invita a observar cuál es el PID del primer proceso iniciado por el usuario justo después del arranque de la máquina. Ronda el 150 en Mac OS, pero es alrededor de 1800 en una distribución clásica. Por lo tanto, hay un problema. El init systemd permite acelerar el arranque de servicios, siempre basado en eventos y de forma asíncrona. Pero va más allá, especialmente anticipándose a las necesidades de varios servicios para acelerar su carga en el futuro, averiguar y gestionar el estado exacto de cada proceso que forme parte de un servicio y tener un control total sobre éstos, sobre todo gracias a una interfaz CLI y otra gráfica. Además, aboga por sustituir el shell script por un lenguaje compilado.

En la guerra de los sucesores de init System V, systemd es el que ha ganado. Sus grandes cualidades y su adopción por el conjunto de los fabricantes para sus distribuciones (Fedora, RHEL 7, OpenSUSE, SLES), y sobre todo la decisión de Debian en febrero de 2014 de adoptar systemd para su versión mayor Jessie, han vencido las últimas resistencias, incluidas las de Canonical, fundador de Ubuntu y autor de upstart. La versión 14.10 será la primera en utilizar systemd.

A pesar de que systemd utiliza mecanismos eficientes, hace invocaciones a funciones (API) únicamente presentes en el núcleo de Linux como, entre otras, el concepto de cgroups que permite agrupar procesos en un mismo grupo (cada proceso asociado a un servicio se ubica en un mismo grupo). Por ello, systemd ya no es compatible con POSIX y, por lo tanto, depende de Linux. A no ser que se contribuya con aportaciones grandes y pesadas a los núcleos de otros Unix, systemd no es portable.

2. Unidades objetivo y servicios

En systemd, el concepto de nivel de ejecución ya sólo existe para la compatibilidad con System V. El estado deseado por el sistema después de haber ejecutado los servicios se llama **unidad objetivo** o**target unit**. El componente básico de systemd es la **unidad** o **unit**. Existen varias: servicios, sockets, periféricos, objetivos, etc. Un objetivo es básicamente el punto de sincronización entre unidades en espera. De este modo, dispondrá, por ejemplo, de unidades objetivo correspondientes a cada nivel de ejecución, pero puede llamarlas como desee.

Las definiciones de las distintas unidades están en **/lib/systemd/system**. Las unidades objetivo o los servicios que deben ser gestionados por el sistema durante el arranque están en**/etc/system/system**, en forma de enlaces simbólicos o copias de los archivos anteriores.

Puede activar o definir varios objetivos al mismo tiempo y no limitarse a uno solo, único y pesado. Por ejemplo, un objetivo contendrá la activación de los servicios de red, otro los servicios de audio y el último para el inicio del entorno gráfico, etc. Un objetivo se puede agrupar en otro.

Servicios

Los servicios acaban con el sufijo **.service**. A continuación veamos un ejemplo con el contenido del servicio ntpd que representa sus dependencias y su comando de ejecución:

```
# cat ntpd.service
[Unit]
Description=Network Time Service
After=syslog.target ntpdate.service

[Service]
EnvironmentFile=/etc/sysconfig/ntpd
ExecStart=/usr/bin/ntpd -n -u ntp:ntp $OPTIONS

[Install]
WantedBy=multi-user.target
```

Una línea interesante es la de **WantedBy**. Sirve para indicar que este servicio es necesario para el objetivo multi-user.target (lo que correspondrá por ejemplo al nivel 3 del init System V). Un objetivo no define qué servicios gestiona, pero cada servicio define para qué unidad es necesario. Por lo tanto, es factible construir un árbol de dependencias.

Objetivos

A continuación tenemos un ejemplo de la unidad **graphical.target**, que corresponde al runlevel 5:

```
# cat graphical.target
[Unit]
Description=Graphical Interface
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

El comando siguiente devuelve la lista de servicios necesarios para este objetivo:

```
# grep -e 'WantedBy.*graphical.target' *
accounts-daemon.service:WantedBy=graphical.target
console-kit-daemon.service:WantedBy=graphical.target
firstboot-graphical.service:WantedBy=graphical.target
rtkit-daemon.service:WantedBy=graphical.target
system-setup-keyboard.service:WantedBy=graphical.target
```

Ahora bien, compruebe también que cada unidad dispone de valores **Requires** o **After**. El objetivo graphical.target requiere que multi-user.target esté parado:

```
# grep -e 'WantedBy.*multi-user.target' *
abrtd.service:WantedBy=multi-user.target
acpid.service:WantedBy=multi-user.target
atd.service:WantedBy=multi-user.target
avahi-daemon.service:WantedBy=multi-user.target
crond.service:WantedBy=multi-user.target
firstboot.service:WantedBy=multi-user.target
irqbalance.service:WantedBy=multi-user.target
mcelog.service:WantedBy=multi-user.target
NetworkManager.service:WantedBy=multi-user.target
NetworkManager-wait-online.service:WantedBy=multi-user.target
ntpdate.service:WantedBy=multi-user.target
ntpd.service:WantedBy=multi-user.target
rdisc.service:WantedBy=multi-user.target
remote-fs.service:WantedBy=multi-user.target
rsyslog.service:WantedBy=multi-user.target
smartd.service:WantedBy=multi-user.target
```

```
sssd.service:WantedBy=multi-user.target
```

Y cada unidad servicio puede contener valores Requires o After... Así, definiendo reglas de dependencia de forma simple (tal servicio depende de tal o cual otro y sólo debe ejecutarse después de tal o cual otro), la gestión de servicios es asíncrona.

3. Acciones

El comando **systemctl** permite controlar la ejecución de servicios. A continuación tenemos algunos ejemplos de uso.

Listado de unidades

Todas las unidades activas:

```
# systemctl
UNIT                        LOAD   ACTIVE SUB          JOB DESCRIPTION
...
-.mount                     loaded active mounted          /
home-seb-.gvfs.mount        loaded active mounted          /home/seb/.gvfs
home.mount                  loaded active mounted          /home
media.mount                 loaded active mounted          Media Directory
sys-fs-f...nections.mount loaded active mounted          /sys/fs/fuse/
connections
tmp.mount                   loaded active mounted          /tmp
var-lib-...c_pipefs.mount loaded active mounted          /var/lib/nfs/
rpc_pipefs
var-tmp.mount               loaded active mounted          /var/tmp
systemd-...-plymouth.path loaded active mounted          Forward Password
Requests to Plymouth Directory Watch
systemd-...word-wall.path loaded active mounted          Forward Password
Requests to Wall Directory Watch
abrt-ccpp.service           loaded active mounted          LSB: Installs
log for oops messages, creates ABRT dump directories for each oops
abrtd.service               loaded active mounted          LSB: Watches system
Bug Reporting Tool
...
```

Todas las unidades:

```
# systemctl --all
...
alsa-restore.service        loaded inactive dead           Restore Sound Card
State
alsa-store.service          loaded inactive dead           Store Sound Card
State
atd.service                 loaded active   running        Job spooling tools
auditd.service              loaded active   running        SYSV: This starts
the Linux Auditing System Daemon, which collects security related events in
a dedicated audit log. If this
avahi.daemon.service        loaded active   running        Avahi mDNS/DNS-SD
Stack
btseed.service              loaded inactive dead           SYSV: Start up seed
clients for BitTorrent
bttrack.service             loaded inactive dead           SYSV: Start up
tracker for BitTorrent
cgconfig.service            loaded inactive dead           LSB: start and
stop the WLM configuration
cgred.service               loaded inactive dead           LSB: Start and
stop the cgroups rules engine daemon
console-...daemon.service loaded active   running        Console Manager
```

```
console-...-start.service loaded active   exited        Console System
Startup Logging
cpuspeed.service          loaded active   exited        LSB: processor
frequency scaling support
crond.service             loaded active   running       Command Scheduler
cups.service              loaded active   running       LSB: The CUPS
scheduler
...
```

Parada y reinicio de unidades

```
# systemctl stop cups.service
# systemctl start cups.service
# systemctl restart cups.service
# systemctl reload cups.service
```

Las versiones recientes de systemd permiten ir más allá del sufijo **.service**.

Cambiar el nivel de ejecución

Los niveles de ejecución se conservan por razones de compatibilidad con System V, pero son reemplazados en la práctica por unidades objetivo del mismo nombre. El nivel de ejecución puede modificarse dinámicamente detallando el nivel objetivo deseado, tal y como se describe a continuación:

```
# systemctl isolate runlevel5.target
```

Para modificar el nivel de ejecución predeterminado, debe crear un enlace simbólico llamado default.target a la unidad objetivo deseada:

```
# ln -sf /lib/systemd/system/runlevel5.target
/etc/systemd/system/default.target
```

Averiguar el nivel de ejecución actual

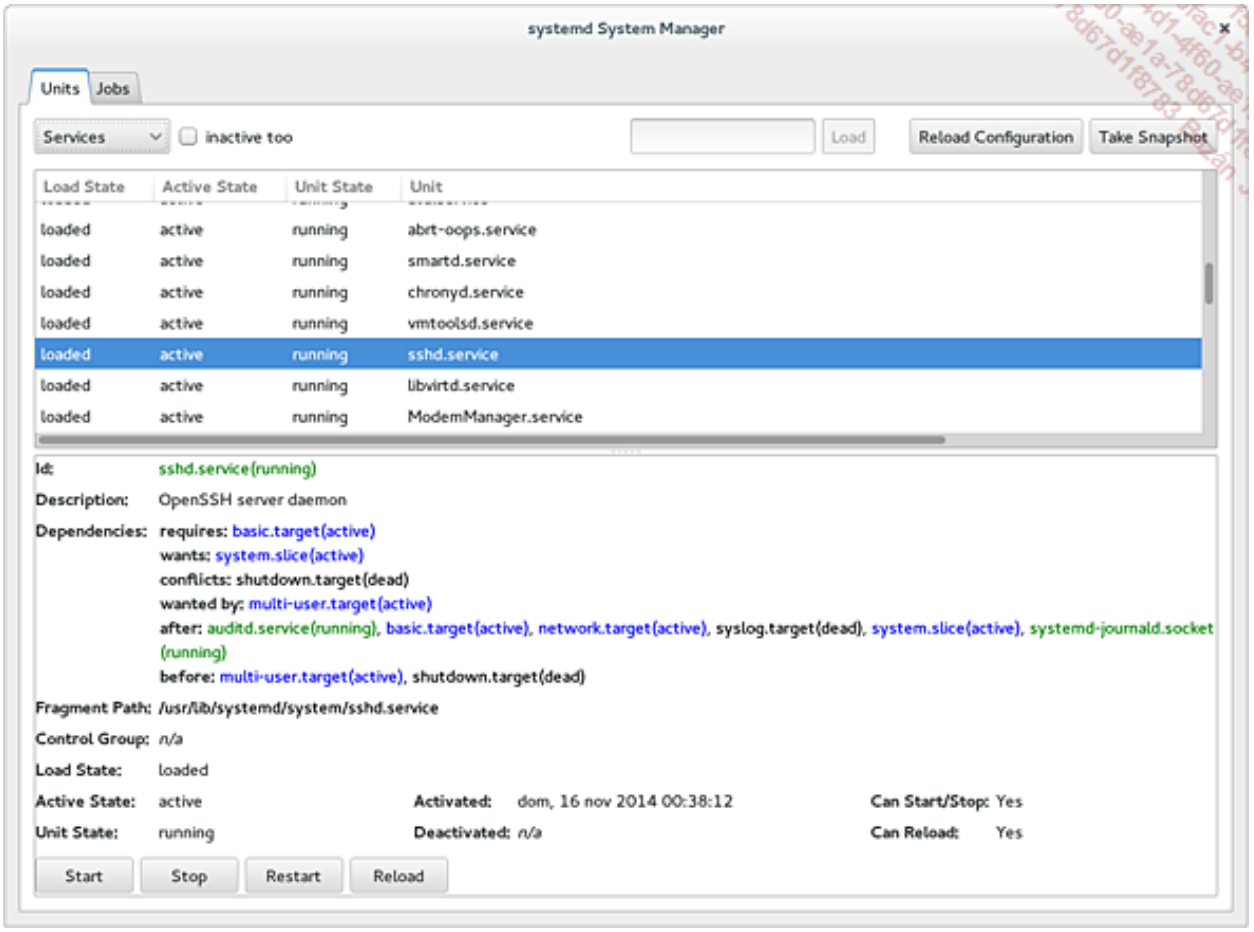
Como cabe la posibilidad de que varias unidades objetivos estén activas a la vez, este concepto no tiene sentido alguno, excepto en los sistemas donde el objetivo es el nivel de ejecución y ningún otro.

El siguiente comando indica cuáles son las unidades objetivo activas:

```
# systemctl list-units -type=target
UNIT                      LOAD   ACTIVE SUB     JOB DESCRIPTION
basic.target              loaded active active      Basic System
cryptsetup.target         loaded active active      Encrypted Volumes
getty.target              loaded active active      Login Prompts
graphical.target          loaded active active      Graphical Interface
local-fs.target           loaded active active      Local File System
mail-tra...r-agent.target loaded active active      Mail Transfer Agent
multi-user.target         loaded active active      Multi-User
network.target            loaded active active      Network
remote-fs.target          loaded active active      Remote File System
sockets.target            loaded active active      Sockets
sound.target              loaded active active      Sound Card
swap.target               loaded active active      Swap
sysinit.target            loaded active active      System Initialization
syslog.target             loaded active active      Syslog
time-sync.target          loaded active active      System Time Synchronized
```

4. Interfaz gráfica

El comando **systemadm** inicia la interfaz gráfica de systemd. En Fedora, está en el paquete **systemd-ui**. Esta interfaz permite controlar el conjunto de unidades systemd y realizar todas las acciones del comando **systemctl** en varias unidades.



Interfaz de control de systemd

upstart

1. Fundamentos

upstart es el sustituto del servicio init, inicialmente desarrollado para Ubuntu por un empleado de la sociedad Canonical, que funciona de manera asíncrona y basado en eventos. En Ubuntu lo encontramos hasta la versión 14.04, en Debian hasta la versión 6, al igual que ciertas versiones de Fedora y RHEL6 y las versiones derivadas asociadas (Oracle Linux por ejemplo). Si bien el cambio a systemd se ha efectuado para estas distribuciones, upstart, simple y ligero, se sigue utilizando en muchos productos derivados.

upstart controla el conjunto de servicios, gestiona su arranque y su detención y supervisa también su funcionamiento. Por razones evidentes, mantiene compatibilidad con init System V: cualquier servicio System V funcionará con upstart, sin embargo los servicios upstart no funcionan con System V. upstart es de momento el init predeterminado de muchas distribuciones, y su migración se efectua con facilidad gracias a esta compatibilidad.

Con upstart, los servicios pueden iniciarse de forma asíncrona y, por lo tanto, pueden paralelizarse y no iniciarse secuencialmente como con init. De este modo, se pueden definir dependencias entre los servicios y optimizar sus inicios y paradas. Basado en eventos, upstart podrá, por ejemplo, iniciar los servicios Bluetooth únicamente si se ha conectado un pendrive de este tipo, y no de forma inútil y sistemática en el arranque de la máquina.

2. Archivos

Los archivos de configuración de upstart están en **/etc/init**. Son archivos de configuración con sintaxis específica, aunque también pueden contener shell script clásico. A continuación tenemos a modo de ejemplo el archivo de configuración que se encarga del servicio sshd:

```
# ssh - OpenBSD Secure Shell Server
#
# The OpenSSH server provides secure shell access to the system.

Description     "OpenSSH server"

start on filesystem or runlevel [2345]
stop on runlevel [!2345]

respawn
respawn limit 10 5
umask 022

pre-start script
test -x /usr/sbin/sshd || { stop; exit 0; }
test -e /etc/ssh/sshd_not_to_be_run && { stop; exit 0; }
test -c /dev/null || { stop; exit 0; }

mkdir -p -m0755 /var/run/sshd
end script

# if you used to set SSHD_OPTS in /etc/default/ssh, you can change the
# 'exec' line here instead
exec /usr/sbin/sshd -D
```

La sintaxis es comprensible para el administrador. Las directivas **start** y **stop** indican con qué eventos se inicia y se detiene el servicio. En este caso, el servicio se iniciará si los sistemas de archivos están montados o si se está en los niveles de 2 a 5. Se detendrá en todos los niveles salvo los que están entre 2 y 5 (ambos incluidos), a saber: 0, 1 y 6. Una parte de la sintaxis retoma la de inittab, especialmente en las acciones (por ejemplo respawn).

3. Nivel por defecto

El archivo **/etc/inittab** ya no es necesario y no se interpretará, pero hay una excepción. Si está presente, es únicamente por la línea **initdefault**. Si no existe, el nivel de ejecución predeterminado se indica en la variable **DEFAULT_RUNLEVEL** del archivo **/etc/init/rc-sysinit.conf**. El nivel de ejecución siempre puede pasarse por parámetro al núcleo en el arranque.

4. Compatibilidad con System V

Los scripts **rc-sysinit.conf**, **rcS.conf** y **rc.conf** siempre están para garantizar la compatibilidad con los scripts System V. El script **rc.conf** es el que va a ejecutar el archivo **/etc/init.d/rc** con el nivel de ejecución pasado por parámetro. A partir de este punto, el método es idéntico al método init System V.

5. Comandos de control

Aunque un script init System V siempre puede controlarse con su nombre o por el comando service, estos métodos no son recomendables. upstart dispone de un cierto número de comandos que los reemplazan.

Iniciar un servicio

Utilice el comando start:

```
# start ufw
ufw start/running
```

Recargar un servicio

Utilice el comando restart:

```
# restart ufw
ufw start/running
```

Parar un servicio

Utilice el comando stop:

```
# stop ufw
ufw stop/waiting
```

Estado de un servicio

Utilice el comando status:

```
# status ufw
ufw start/running
```

initctl

Puede utilizar el comando **initctl** para controlar servicios. Con él se pueden realizar todas las acciones posibles. Recibe como primer parámetro una acción y, si procede, un segundo parámetro con el nombre del servicio asociado. Por ejemplo, puede listar los servicios:

```
# initctl list
avahi-daemon start/running, process 1410
mountall-net stop/waiting
nmbd start/running, process 2030
qemu-kvm start/running
rc stop/waiting
rpc_pipefs (idmapd) start/running
rsyslog start/running, process 1362
screen-cleanup stop/waiting
tty4 start/running, process 1511
udev start/running, process 490
upstart-udev-bridge start/running, process 486
ureadahead-other stop/waiting
apport stop/waiting
console-setup stop/waiting
...
```

El comando también permite emitir un evento con **emit**. Por ejemplo, para cambiar el nivel de ejecución (lo que siempre se puede hacer con los comandos telinit, halt, reboot o shutdown):

```
# initctl emit runlevel RUNLEVEL=2
```

6. Activación y desactivación de un servicio

No hay sustituto hasta la fecha de **chkconfig** o **update-rc.d** para los servicios upstart. Sin embargo, si la distribución mantiene cierta compatibilidad con System V y upstart sirve básicamente para controlar estos servicios, estos comandos continuarán siendo funcionales para cualquier servicio albergado en **/etc/init.d**.

En caso contrario, necesitará:

- detener manualmente el servicio,
- modificar los eventos start y stop en los servicios asociados en /etc/init.

Consultar el registro del sistema

1. dmesg

El comando **dmesg** permite recuperar los mensajes del núcleo emitidos al arrancar la máquina y los emitidos después. El registro de dmesg es circular. Después de cierto número de mensajes, los primeros desaparecen. Sin embargo, estas entradas en el registro no están perdidas, ya que el servicio syslogd (capítulo La red) las escribe en archivos.

El administrador, ingeniero o usuario del sistema Linux suele iniciar este comando para comprobar la presencia de posibles errores. En efecto, después del boot, los mensajes siguen llegando, en particular durante la conexión en caliente de periféricos, durante la carga de algunos módulos, cuando se producen averías, durante una corrupción del sistema de archivos, etc.

Se ha truncado voluntariamente el ejemplo siguiente a las primeras líneas, ya que la salida original contiene más de 500. Las primeras muestran todo el principio de la ejecución del núcleo (información facilitada por la BIOS). En medio se muestra la detección del primer disco duro y de sus particiones. El final muestra lo que ocurre al insertar de un pendrive, después del boot, durante una utilización normal y en el momento de desconectarse.

```
# dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.13.0-24-generic (buildd@batsu) (gcc version 4.8.2
(Ubuntu 4.8.2-19ubuntu1) ) #47-Ubuntu S
MP Fri May 2 23:30:00 UTC 2014 (Ubuntu 3.13.0-24.47-generic 3.13.9)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.13.0-24-generic
root=UUID=3ef2d0d1-ba58-4fd0-b578-2f32a1102455 r
o
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Centaur CentaurHauls
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000007ffeffff] usable
[    0.000000] BIOS-e820: [mem 0x000000007fff0000-0x000000007fffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.5 present.
[    0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[    0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[    0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[    0.000000] No AGP bridge found
... (y así hasta el final)
```

El resultado no es muy explícito. Podemos mejorar la visualización temporal para que sea más legible empleando -T.

```
# dmesg -T
[lun dic  1 11:19:45 2014] Base memory trampoline at [ffff88000009a000] 9a000 size 20480
[lun dic  1 11:19:45 2014] init_memory_mapping: 0000000000000000-00000000c0000000
[lun dic  1 11:19:45 2014]  0000000000 - 00c0000000 page 2M
[lun dic  1 11:19:45 2014] kernel direct mapping tables up to c0000000 @ 1fffc000-20000000
[lun dic  1 11:19:45 2014] init_memory_mapping: 0000000100000000-0000000140000000
[lun dic  1 11:19:45 2014]  0100000000 - 0140000000 page 2M
[lun dic  1 11:19:45 2014] kernel direct mapping tables up to 140000000 @ bfffe000-c0000000
```

```
[lun dic  1 11:19:45 2014] RAMDISK: 37a88000 - 37d3c000
[lun dic  1 11:19:45 2014] ACPI: RSDP 00000000000f6b80 00024 (v02 PTLTD )
[lun dic  1 11:19:45 2014] ACPI: XSDT 00000000bfeedf22 0005C (v01 INTEL  440BX    06040000
VMW  01324272)
[lun dic  1 11:19:45 2014] ACPI: FACP 00000000bfefee98 000F4 (v04 INTEL  440BX    06040000
PTL  000F4240)
[lun dic  1 11:19:45 2014] ACPI: DSDT 00000000bfeee910 10588 (v01 PTLTD  Custom   06040000
MSFT 03000001)
[lun dic  1 11:19:45 2014] ACPI: FACS 00000000bfefffc0 00040
[lun dic  1 11:19:45 2014] ACPI: BOOT 00000000bfeee8e8 00028 (v01 PTLTD  $SBFTBL$ 06040000
LTP 00000001)
[lun dic  1 11:19:45 2014] ACPI: APIC 00000000bfeee526 003C2 (v01 PTLTD  ? APIC   06040000
LTP 00000000)
```

También puede filtrar únicamente los mensajes del núcleo y el nivel de advertencia de la siguiente manera:

```
# dmesg -k -lerr
[    4.041379] piix4_smbus 0000:00:07.0: SMBus base address uninitialized -
upgrade BIOS or use force_addr=0xaddr
[    4.975956] intel_rapl: no valid rapl domains found in package 0
```

Para sacarle partido al resultado, lo ideal sería o bien redireccionarlo a un archivo para un análisis más en frío, o bien utilizar el comando **grep** a propósito, si sabe lo que busca.

```
# dmesg|grep CPU
[    0.000000] CPU MTRRs all blank - virtualized system.
[    0.000000] ACPI: SSDT 000000007fff02a0 0001CC (v01 VBOX   VBOXCPUT 00000002
INTL 20100528)
[    0.000000] smpboot: Allowing 1 CPUs, 0 hotplug CPUs
[    0.000000] setup_percpu: NR_CPUS:256 nr_cpumask_bits:256 nr_cpu_ids:1
nr_node_ids:1
[    0.000000] PERCPU: Embedded 29 pages/cpu @ffff88007fc00000 s86336 r8192 d24256
u2097152
[    0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
[    0.000000]     RCU restricting CPUs from NR_CPUS=256 to nr_cpu_ids=1.
[    0.000000]     Offload RCU callbacks from all CPUs
[    0.000000]     Offload RCU callbacks from CPUs: 0.
[    0.006379] mce: CPU supports 0 MCE banks
[    0.101196] smpboot: CPU0: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (fam: 06,
model: 3a, stepping: 09)
[    0.110294] Performance Events: unsupported p6 CPU model 58 no PMU driver,
software events only.
[    0.112094] x86: Booted up 1 node, 1 CPUs
[    0.477696] microcode: CPU0 sig=0x306a9, pf=0x2, revision=0x19
[    0.622672] ledtrig-cpu: registered to indicate activity on CPUs
```

2. /var/log/messages

Sea cual sea la distribución empleada, **/var/log/messages** es el archivo central de los mensajes del sistema, provengan del núcleo o de los servicios. El contenido de este archivo, gestionado por syslog, refleja el estado global del sistema (y no únicamente del núcleo) durante su utilización. En un sistema clásico, su contenido retoma el procedente del comando **dmesg** y el de varios servicios.

Las líneas tienen una firma temporal. Sin que se produzca alguna acción especial (ver logrotate en el capítulo La red), el archivo crece con el tiempo y no se purga. Un archivo de mensajes puede contener varios miles de líneas, ¡sobre todo si se producen problemas!

```
# wc -l < messages
8453
```

Así, como con el comando **dmesg**, no olvide efectuar un grep para seleccionar sus líneas (o tail, head,

etc.).

```
# tail -100 /var/log/messages | grep fglrx
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] interrupt source 20008000
successfully disabled!
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] enable ID = 0x00000000
Feb  5 09:12:22 p64p17bicb3 kernel: [fglrx] Receive disable interrupt
message with irqEnableMask: 20008000; dwIRQEnableId: 00000004
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Maximum main memory to use
for locked dma buffers: 867 MBytes.
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] GART Table is not in FRAME_
BUFFER range
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Reserve Block - 0 offset =
0X0 length = 0X40000
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Reserve Block - 1 offset =
0X3ff5000 length = 0Xb000
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] interrupt source 20008000
successfully enabled
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] enable ID = 0x00000004
Feb  5 09:13:32 p64p17bicb3 kernel: [fglrx] Receive enable interrupt
message with irqEnableMask: 20008000
```

En algunas distribuciones el archivo messages puede estar vacío. Eso significará que el comando syslog, del que hay una descripción más adelante en este capítulo, ha distribuido los mensajes asociados en otros archivos.

3. /var/log/syslog

En las distribuciones de tipo Debian o Ubuntu, el archivo central de los mensajes del sistema es**/var/log/syslog**. Lo explicado en el punto anterior es igual a lo que se reporta en este archivo. La opción de un archivo u otro depende solamente de la configuración del servicio syslog (o rsyslog, o syslog-ng).

Servicios y módulos del núcleo

1. Presentación

El núcleo es el corazón del sistema operativo Linux. Linux es el núcleo, el sistema operativo tal y como lo desarrolló originalmente Linus Torvalds. El sistema operativo Linux en su amplio contexto se compone del núcleo y de las herramientas operativas básicas, las herramientas GNU. De igual forma, varios son los que dicen que Linux debería llamarse GNU/Linux: GNU por las herramientas, Linux por el núcleo, GNU/Linux para el Sistema Operativo.

El núcleo de Linux es libre. Sus fuentes están disponibles. Por lo tanto, se puede volver a compilar para adaptarlo específicamente a sus necesidades, modificarlo y añadirle extensiones.

El núcleo de Linux forma parte de la familia de los núcleos monolíticos, es decir, que agrupan todas sus funcionalidades y componentes en un programa único. Sin embargo, desde la versión 2.0 (o más bien la versión de desarrollo 1.3 para ser más precisos) el núcleo es modular.

El núcleo se llama **kernel**. Está en **/boot** y su nombre, por convención, comienza a menudo por**vmlinuz-X.Y.Z.p-Vtxt**.

Se obtiene la versión del núcleo con el comando **uname**.

```
$ uname -r
2.6.38-8-generic
```

Las letras tienen un significado particular.

- **X**: versión mayor del núcleo. Entre la versión 1 y 2, el paso al funcionamiento modular ha sido determinante, así como la nueva implementación de la capa de red. La versión 3 vio la luz en verano de 2011.
- **Y**: un valor par representa una rama estable del núcleo. Una versión impar representa una rama de desarrollo (¡cuidado!). Cada incremento par (0,2,4,6) representa una evolución importante del núcleo.

Las versiones 2.6 y 3.x no disponen de ramas de desarrollo, ya que evolucionan demasiado rápido. Los desarrolladores decidieron implementar sus novedades directamente en la versión estable.

- **Z**: versión menor del núcleo. Cuando un lote de modificaciones, con respecto a una versión anterior, necesita la difusión du nuevo núcleo, entonces se incrementa esta cifra. Por ejemplo, un lote que agrupa una modificación del sistema de sonido (Alsa, que pasa de 1.0.8 a 1.0.9), del sistema de archivos (se añade de ReiserFS 4), y así sucesivamente...
- **p**: versión corregida o intermedia presente desde la versión 2.6. Cuando el núcleo necesita una actualización menor (corrección de uno o dos bugs, etc.) pero pocas funcionalidades nuevas, se incrementa este valor.
- **V**: al igual que con los paquetes, versión propia del editor de la distribución.
- **txt**: a veces se añade un texto para dar precisiones sobre el núcleo. Por ejemplo, smp indica un núcleo multiprocesador.

La versión 3.0 salió en verano de 2011. Aunque sea una etapa muy importante, la numeración en la versión 3.0 sólo es una decisión que se podría catalogar como política, incluso festiva. Entre las versiones del núcleo 2.6.0 y 2.6.39 han transcurrido bastantes años y ha habido un gran número de modificaciones que han provocado que entre ambas versiones haya poco en común. Como el núcleo 2.6 no ha tenido una rama de desarrollo, la diferencia entre las versiones 2.6.39 y 3.0 no serán tan importantes como en las versiones anteriores. ¡Son sobre todo los 20 años de Linux los que han

precipitado el cambio de versión! Linus Torvalds presenta así la versión 3.0:

"Entonces, ¿cuáles son los cambios más importantes? NINGUNO. Absolutamente ninguno. Por supuesto, tenemos las habituales modificaciones de dos o tres drivers y una gran cantidad de correcciones, pero la versión 3.0 no es más que un simple cambio de numeración, no como con KDE-4 o GNOME-3. La compatibilidad no se ha perdido, ya que no hay nuevas funcionalidades que asusten ni nada parecido. A partir de ahora, se publican nuevas versiones basándose en las fechas y no en las novedades funcionales, tal y como se ha estado haciendo en mucho tiempo. Si quiere una excusa para esta renumeración, digamos que ha sido por los 20 años de Linux."

Por lo tanto, el gran cambio sólo concierne al nuevo modo de numeración.

2. uname

El comando **uname** (unix name) permite obtener toda la información sobre la versión de Unix (de Linux en este caso) de manera precisa y completa.

Parámetro	Resultado
`-m` (máquina)	Tipo de hardware de la máquina.
`-n` (nodename)	Nombre de la máquina.
`-r` (release)	Versión (número) del núcleo.
`-s` (system name)	Nombre del sistema operativo. Por defecto.
`-p` (processor)	Tipo de procesador.
`-i`	Plataforma física.
`-v` (version)	Versión del sistema.
`-a` (all)	Toda la información.

```
$ uname
Linux
$ uname -m
X86_64
$ uname -n
slyserver
$ uname -r
3.13.0-24-generic
$ uname -s
Linux
$ uname -p
$ uname -i
X86_64
$ uname -o
GNU/Linux
$ uname -v
#47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014
$ uname -a
Linux slyserver 3.13.0-24-generic #4-Ubuntu SMP Fri May 2 23:30:00 UTC 2014
x86_64 intel x86_64 GNU/Linux
```

3. Gestión de los módulos

Los componentes básicos (gestor de tareas, gestión de la memoria, de los procesos, API, etc.) están siempre dentro de un programa único. Pero algunos drivers de periférico, sistemas de archivos, extensiones, protocolos de redes, etc., pueden estar presentes en forma de módulos. Los módulos se comunican con el núcleo a través de un API común. Se ejecutan en el espacio del núcleo. Se pueden configurar. Se pueden cargar y descargar a petición y así evitar un reinicio de la máquina. El

añadido de un nuevo módulo (desde sus fuentes por ejemplo) no necesita reinicio.

Los módulos están en **/lib/modules/$(uname -r)**.

```
# cd /lib/modules/$(uname -r)
# pwd
/lib/modules/3.13.0-24-generic
```

Los módulos tienen un nombre que termina por "ko" para kernel object. La terminación de origen era "o" para los núcleos 2.0 a 2.4. Efectivamente, es eso: archivos de objetos relacionados (linked) de manera dinámica al núcleo, que proponen así una API adicional.

```
# cd /lib/modules/$(uname -r)/kernel/fs/vfat
# file msdos.ko
msdos.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
```

La palabra "relocatable" indica que está ante un archivo de objeto. Los núcleos recientes adjuntan un valor suplementario, el BuildID.

a. lsmod

El comando **lsmod** lista los módulos actualmente cargados, con sus posibles dependencias.

```
# lsmod
Module                  Size  Used by
snd_intel8x0           38153  0
snd_ac97_codec        130285  1 snd_intel8x0
ac97_bus               12730  1 snd_ac97_codec
snd_pcm               102099  2 snd_ac97_codec,snd_intel8x0
serio_raw              13462  0
snd_page_alloc         18710  2 snd_intel8x0,snd_pcm
usbhid                 52616  0
hid                   106148  2 hid_generic,usbhid
snd_timer              29482  1 snd_pcm
i2c_piix4              22155  0
parport_pc             32701  0
snd                    69238  4 snd_ac97_codec,snd_intel8x0,snd_timer,snd_pcm
soundcore              12680  1 snd
lp                     17759  0
parport                42348  3 lp,ppdev,parport_pc
mac_hid                13205  0
psmouse               102222  0
ahci                   25819  2
libahci                32168  1 ahci
e1000                 145174  0
...
...
```

La primera columna indica el nombre del módulo cargado. Su nombre refleja a menudo para qué sirve. La segunda columna da el tamaño del módulo. La tercera columna suministra un contador de utilización (cuántos componentes del sistema acceden a los módulos). La última columna proporciona la lista de los módulos que utilizan (o, por lo tanto, dependen) del primero.

En el ejemplo anterior, el módulo lp utiliza el módulo parport.

lsmod sólo vuelve a dar forma al contenido del archivo virtual **/proc/modules**.

```
# cat /proc/modules
snd_intel8x0 38153 0 - Live 0xffffffffa0121000
snd_ac97_codec 130285 1 snd_intel8x0, Live 0xffffffffa00fc000
```

```
ac97_bus 12730 1 snd_ac97_codec, Live 0xffffffffa002a000
snd_pcm 102099 2 snd_intel8x0,snd_ac97_codec, Live 0xffffffffa00e2000
serio_raw 13462 0 - Live 0xffffffffa00dd000
snd_page_alloc 18710 2 snd_intel8x0,snd_pcm, Live 0xffffffffa00c2000
usbhid 52616 0 - Live 0xffffffffa00c9000
hid 106148 2 hid_generic,usbhid, Live 0xffffffffa00a7000
snd_timer 29482 1 snd_pcm, Live 0xffffffffa0099000
i2c_piix4 22155 0 - Live 0xffffffffa0086000
parport_pc 32701 0 - Live 0xffffffffa0090000
snd 69238 4 snd_intel8x0,snd_ac97_codec,snd_pcm,snd_timer, Live 0xffffffffa0074000
soundcore 12680 1 snd, Live 0xffffffffa0038000
lp 17759 0 - Live 0xffffffffa0045000
parport 42348 3 ppdev,parport_pc,lp, Live 0xffffffffa0065000
mac_hid 13205 0 - Live 0xffffffffa0025000
psmouse 102222 0 - Live 0xffffffffa004b000
ahci 25819 2 - Live 0xffffffffa003d000
libahci 32168 1 ahci, Live 0xffffffffa002f000
e1000 145174 0 - Live 0xffffffffa0000000
...
```

b. depmod

El comando **depmod** actualiza el árbol de las dependencias entre los módulos modificando el archivo **`modules.dep`**.

El archivo **`/lib/modules/$(uname -r)/modules.dep`** contiene dos columnas. La primera columna es la ruta del módulo, la segunda es la lista de dependencias: los módulos que se deben cargar también para que el primero funcione. Aquí, tenemos el ejemplo de la línea que corresponde al módulo `lp`:

```
$ grep "/lp.ko:" modules.dep
kernel/drivers/char/lp.ko: kernel /drivers/parport/parport.ko
```

El módulo `lp.ko` depende del módulo `parport.ko`. Por lo tanto se debe cargar el módulo`parport.ko` primero para que `lp.ko` funcione.

El uso más corriente de **depmod** incluye el parámetro `-a`, que vuelve a construir las dependencias de todos los módulos que corresponden al núcleo actual. Se ejecuta esta acción a cada inicio del sistema, pero si compila o instala nuevos módulos, debe reiniciar manualmente este comando para tener en cuenta las nuevas dependencias.

```
# depmod -a
```

c. modinfo

El comando **modinfo** ofrece toda la información necesaria sobre un módulo:

- nombre del archivo correspondiente;
- descripción del módulo;
- autor;
- licencia;
- dependencias;
- parámetros;
- alias.

Los módulos no proporcionan toda esta información. Algunos módulos no tienen parámetros. En el primer ejemplo, el módulo `parport` no tiene parámetros. Éstos se instalan como opciones de montaje.

```
# modinfo parport
filename:       /lib/modules/3.13.0-24-generic/kernel/drivers/parport/parport.ko
license:        GPL
srcversion:     278BD67C796634644178740
depends:
intree:         Y
vermagic:       3.13.0-24-generic SMP mod_unload modversions
signer:         Magrathea: Glacier signing key
sig_key:        69:B0:D0:A7:9B:85:D9:06:21:70:6E:8D:06:60:4D:73:0B:35:9F:C0
sig_hashalgo:   sha512
```

En este segundo ejemplo, el módulo dispone de parámetros. Este módulo permite utilizar una webcam que contiene un chip compatible con el módulo uvcvideo. Los parámetros están truncados de manera voluntaria.

```
# modinfo uvcvideo
filename:       /lib/modules/3.13.0-24-generic/kernel/drivers/media/usb/uvc/uvcvideo.ko
version:        1.1.1
license:        GPL
description:    USB Video Class driver
author:         Laurent Pinchart <laurent.pinchart@ideasonboard.com>
srcversion:     522EC7538EA0B6E2E671A4F
alias:          usb:v*p*d*dc*dsc*dp*ic0Eisc01ip00in*
alias:          usb:v045Ep00F8d*dc*dsc*dp*ic0Eisc01ip00in*
alias:          usb:v0458p706Ed*dc*dsc*dp*ic0Eisc01ip00in*
alias:          usb:v0416pA91Ad*dc*dsc*dp*ic0Eisc01ip00in*
...
depends:        videodev,videobuf2-core,videobuf2-vmalloc
intree:         Y
vermagic:       3.13.0-24-generic SMP mod_unload modversions
signer:         Magrathea: Glacier signing key
sig_key:        69:B0:D0:A7:9B:85:D9:06:21:70:6E:8D:06:60:4D:73:0B:35:9F:C0
sig_hashalgo:   sha512
parm:           clock:Video buffers timestamp clock
parm:           nodrop:Don't drop incomplete frames (uint)
parm:           quirks:Forced device quirks (uint)
parm:           trace:Trace level bitmask (uint)
parm:           timeout:Streaming control requests timeout (uint)
```

Un mismo módulo puede gestionar varios tipos de componentes físicos. Existen varias webcams que disponen de un chip ov511 o similar que se pueden gestionar con el módulo uvcvideo. Para que el núcleo sepa qué módulo cargar durante la detección de la webcam, o para que el módulo sepa qué tipo de componente físico debe gestionar durante su carga, dispone de alias de hardware, usb, pci, scsi, etc., que permiten reconocer los periféricos que debe gestionar.

Se pasan los parámetros al módulo mediante los comandos **insmod** o **modprobe**, con el archivo **`/etc/modprobe.conf`** o los que están en **`/etc/modprobe.d`**.

d. insmod

El comando **insmod** carga un módulo dado sin gestionar las dependencias. Coge como parámetro un nombre de módulo con su ruta. Le corresponde a usted gestionar el orden de carga de los módulos para evitar errores relacionados con símbolos no resueltos causados por un problema de dependencia.

En el ejemplo siguiente, como los módulos parport y lp no están cargados, se intenta cargar el módulo `lp.ko` solo. Se produce un error, ya que este módulo depende sólo de `parport.ko`. El retorno de **dmesg** es elocuente al respecto.

```
# lsmod|grep lp
```

```
# ls
lp.ko
# insmod lp.ko
insmod: ERROR: could not insert module lp.ko: Unknown symbol in module
# dmesg | tail -20
...
[  840.533064] lp: Unknown symbol parport_set_timeout (err 0)
[  840.533073] lp: Unknown symbol parport_read (err 0)
[  840.533081] lp: Unknown symbol parport_register_device (err 0)
[  840.533086] lp: Unknown symbol parport_claim_or_block (err 0)
[  840.533092] lp: Unknown symbol parport_register_driver (err 0)
[  840.533097] lp: Unknown symbol parport_release (err 0)
[  840.533111] lp: Unknown symbol parport_unregister_device (err 0)
[  840.533122] lp: Unknown symbol parport_unregister_driver (err 0)
[  840.533135] lp: Unknown symbol parport_negotiate (err 0)
```

En este segundo ejemplo, primero se carga el módulo fat, luego el módulo vfat. No hay error, ya que todas las dependencias están presentes.

```
# pwd
/lib/modules/3.13.0-24-generic/kernel/drivers/parport
# insmod parport.ko
# cd ../char
# insmod lp.ko
# lsmod|grep lp
lp                      17759  0
parport                 42348  1 lp
```

Debería considerar el parámetro `-k`, que permite la limpieza automática de los módulos. Esta opción solicita al sistema que descargue automáticamente un módulo (contador a cero) si ya no se utiliza. Esto permite ganar algunos recursos y obtener un sistema más limpio.

Para transmitir parámetros al módulo, indíquelos a continuación del comando. Por ejemplo, para el módulo uvcvideo, desea que el driver muestre más información:

```
# insmod uvcvideo.ko trace=5
```

e. rmmod

El comando **rmmod** descarga el módulo dado. Se trata del contrario de **insmod** y, al igual que éste, rmmod no gestiona las dependencias:

- No es posible descargar un módulo en curso de utilización.
- No es posible descargar un módulo si otro módulo lo utiliza, incluso si este último no es utilizado (problema de dependencia).

```
# rmmod parport
rmmod: ERROR: Module parport is in use by: lp
```

En este segundo ejemplo, se descarga el módulo lp, luego el módulo parport, en este orden.

```
# rmmod lp
# rmmod parport
```

f. modprobe

El comando **modprobe** carga el módulo dado, así como todas sus dependencias y parámetros contenidos en **`/etc/modprobe.conf`** o los archivos **`/etc/modprobe.d/*`**. El parámetro `-r`permite descargar un módulo y los que dependen de él (si no están en uso).

Cargar el módulo lp mediante **modprobe** significa en la práctica cargar también el módulo parport.

```
# lsmod|grep lp
# modprobe lp
# lsmod|grep lp
lp                  17759  0
parport             42348  1 lp
```

Ahora, como ve que sólo lp utiliza parport (contador a 1), pero que nada utiliza lp (contador a cero), puede intentar descargar lp y los módulos de los cuales depende, si no se usan.

```
# modprobe -r lp
# lsmod|grep lp
```

Puede pasar opciones al módulo; en ese caso, debe comprobar si una línea "install" que corresponde al módulo existe en el archivo modprobe.conf y modificarla para que acepte los parámetros, o utilizar una línea "opciones".

g. modprobe.conf y modprobe.d

¡Ojo! El uso del archivo modprobe.conf ha caído en desuso en las versiones recientes. Cuando está presente, aparece un aviso. En algunas distribuciones recientes, este archivo se ignorará completamente. La configuración se traslada a los archivos independientes albergados en **/etc/modprobe.d**.

La configuración de los módulos está en **/etc/modprobe.conf** y en los archivos albergados en**/etc/modprobe.d/*.conf**. Aunque las distribuciones actuales han dejado de utilizar**modprobe.conf**, sigue siendo importante, ya que todavía se utiliza en distribuciones empresariales, especialmente en las distribuciones anteriores de SLES, RHEL y Debian. En estos archivos la sintaxis es idéntica. Puede definir alias de módulos (muy práctico para las tarjetas de red), pasar opciones a los módulos así como añadir acciones en la carga y descarga de un módulo.

Alias y opciones

Pongamos un ejemplo: tiene dos tarjetas de red. El driver de la primera tarjeta está en el módulo**e1000**. Quiere que se cargue utilizando el nombre **eth0**. El driver de la segunda tarjeta está en el módulo **airo** y desea cargarlo con el nombre **eth1**. También va a especificar parámetros en el módulo uvcvideo, cuya información ha recuperado con **modinfo**.

```
# cat /etc/modprobe.d/network-interfaces.conf
...
alias eth0 e1000
alias eth1 airo
# cat /etc/modprobe.d/webcam.conf
options uvcvideo led=2 compress=1
...
```

Luego carga los módulos por nombre o por alias. En todos los casos, se tienen en cuenta los parámetros automáticamente.

```
# modprobe eth0
# modprobe eth1
# modprobe uvcvideo
```

Se pueden utilizar las opciones de módulos con los alias.

```
alias webcam uvcvideo
options webcam led=2 compress=1
...
```

install y remove

Los comandos **install** y **remove** del archivo **modprobe.conf** son los más potentes. Durante la carga de un módulo, si **modprobe** encuentra un comando **install** asociado, no carga este módulo, sino que ejecuta los comandos indicados. El comando puede ser lo que desee, como por ejemplo la carga de un módulo más adaptado, la instalación de una configuración específica, la ejecución de un script, etc.

El ejemplo siguiente intenta cargar el módulo `uvcvideo_new` si existe, y si no, conmuta a uvcvideo cuando se invoca modprobe webcam.

```
install webcam /sbin/modprobe uvcvideo_new || /sbin/modprobe uvcvideo
```

modprobe puede coger un parámetro práctico que le permita ignorar las líneas install de modprobe.conf: **--ignore-install**. El ejemplo siguiente carga el módulo ahci, luego el módulo`ata_piix` cuando se intenta cargar este último. Sin el parámetro, **modprobe** se ejecutaría en bucle, leyendo e interpretando la línea install a cada intento de carga de `ata_piix`.

```
install ata_piix /sbin/modprobe ahci 2>&1 |:; /sbin/modprobe
--ignore-install ata_piix
```

El comando **remove** hace lo mismo, pero en el momento de descargar el módulo con **modprobe -r**.

CMDLINE_OPTS

Si hay una línea install en modprobe.conf e intenta cargar el módulo correspondiente con modprobe y unos parámetros, no se tendrán en cuenta estos últimos excepto si ha añadido detrás del módulo la cadena $CMDLINE_OPTS.

```
install webcam /sbin/modprobe ov511_new $CMDLINE_OPTS ||
/sbin/modprobe ov511 $CMDLINE_OPTS
```

4. Carga de los módulos al inicio

a. initrd

Algunos módulos pueden ser necesarios en el inicio de la máquina, en particular para montar un sistema de archivos. ¿Cómo montar la partición raíz en ext4 cuando el módulo que gestiona este tipo de sistema de archivos está en esta partición (y no en el núcleo)? Se colocan estos módulos en forma de imagen de disco virtual o **initrd** (inicial ramdisk). Se cargan estos archivos comprimidos al inicio en memoria y son ramdisks. Contienen instrucciones y módulos que se cargan al inicio. Generalmente hay un initrd por núcleo presente.

```
# ls -l /boot/initrd.img-*
-rw-r--r-- 1 root root 19952491 nov 25 22:09 /boot/initrd.img-3.13.0-32-generic
```

Se puede extraer el contenido de **initrd** e incluso modificarlo y volver a construirlo para adaptarlo. Este archivo es una carpeta cpio comprimida con gzip.

```
# zcat /boot/initrd.img-3.13.0-32-generic |cpio -id --no-absolute-filenames
103092 blocks
# ls -l
```

```
total 40
drwxr-xr-x  2 root root 4096 2011-07-18 22:07 bin
drwxr-xr-x  3 root root 4096 2011-07-18 22:07 conf
drwxr-xr-x  8 root root 4096 2011-07-18 22:07 etc
-rwxr-xr-x  1 root root 4096 2011-07-18 22:07 init
drwxr-xr-x  8 root root 4096 2011-07-18 22:07 lib
drwxr-xr-x  2 root root 4096 2011-07-18 22:07 lib64
drwxr-xr-x  2 root root 4096 2011-07-18 22:07 sbin
drwxr-xr-x 10 root root 4096 2011-07-18 22:07 scripts
drwxr-xr-x  4 root root 4096 2011-07-18 22:07 usr
```

Cuando se carga y monta **initrd**, el núcleo intenta ejecutar el script init presente en la raíz del seudosistema de archivos. Es el encargado de cargar los módulos necesarios y establecer la primera configuración básica completa, antes de que el núcleo monte el sistema de archivos raíz y ejecute `/sbin/init`. En la práctica, este archivo (o uno invocado por éste) es el encargado de montar el sistema de archivos raíz y de pasarle el control a `/sbin/init`.

En algunas distribuciones, en particular las antiguas, se invoca a un script llamado `/sbin/init`. Se ha suprimido esto para evitar confundir el script `/sbin/init` de initrd con el "verdadero" init del sistema. Si un archivo **linuxrc** está en la raíz del initrd, se ejecuta. En caso de duda en cuanto al nombre del script iniciado, se puede ver en las fuentes del núcleo `init/main.c`, variable `ram_execute_command`.

```
$ cd /usr/src/linux
$ grep -n ram_execute_command init/main.c
...
865:            ramdisk_execute_command = "/init";
...
```

El comando **mkinitrd** permite volver a construir un archivo initrd estándar, pero a menudo es posible, en función de la distribución, facilitarle una lista de módulos para colocarlos en él, y así cargarlos:

- En Red Hat, se puede utilizar varias veces la opción **--preload=module**, que específica cuáles son los módulos que se deben cargar.
- En openSUSE, le corresponde a la opción **-m "module1 module2 etc."** especificar esta lista. Veremos el método con la utilización de las variables de sysconfig.
- En Debian, hay que colocar los nombres de los módulos en el archivo **/etc/mkinitrd/modules**.

Tenemos el resultado siguiente para un comando **mkinitrd** en openSUSE. La línea "Kernel Modules" especifica los módulos cargados por **initrd**.

```
# mkinitrd

Kernel image:   /boot/vmlinuz-2.6.22.17-0.1-default
Initrd image:   /boot/initrd-2.6.22.17-0.1-default
Root device:    /dev/disk/by-id/scsi-SATA_ST380011A_5JVTH798-part6
(/dev/sda6) (mounted on / as ext3)
Resume device:  /dev/sda5
Kernel Modules: processor thermal scsi_mod libata sata_sil
pata_atiixp fan jbd mbcache ext3 edd sd_mod usbcore ohci-hcd uhci-hcd
ehci-hcd ff-memless hid usbhid
Features:       block usb resume.userspace resume.kernel
Bootsplash:     SuSE (1280x1024)
17588 blocks
```

Un initrd construido correctamente (o más bien un comando **mkinitrd** correcto) recupera también la

salida del comando **modprobe** y el archivo de configuración **modprobe.conf**, que puede (y debe) contener los parámetros de los módulos si es necesario.

Las últimas versiones de Debian y Ubuntu ya no utilizan **mkinitrd**, sino **mkinitramfs**. El principio es el mismo. **initramfs** dispone de su propia configuración situada en **/etc/initramfs-tools/***. Los módulos que se cargarán en el arranque se especifican en **/etc/initramfs-tools/modules**. Para crear un **initrd** para el núcleo actual, utilice el comando siguiente:

```
# mkinitramfs -o /tmp/initrd-$(uname -r)
# ls -l /tmp/initrd-3.13.0-32-generic
-rw-r—r-- 1 root root 19812003 2011-07-18 22:16 /tmp/
initrd-3.13.0-32-generic
```

b. Red Hat: /etc/rc.modules

El método preferido en Red Hat consiste en cargar los módulos desde el **initrd**. Sin embargo, tiene la posibilidad crear un script **/etc/rc.modules** que contendrá los comandos necesarios para la carga de los módulos, principalmente con **modprobe**, como vimos antes. Se ejecuta este archivo con rc.sysinit al iniciar el sistema, mediante **inittab**. Debe ser ejecutable.

c. openSUSE: /etc/sysconfig/kernel

La distribución openSUSE coloca su configuración en un directorio **/etc/sysconfig**. El archivo**/etc/sysconfig/kernel** contiene unos elementos de configuración del núcleo, pero, sobre todo dos variables encargadas de especificar los módulos que se deben cargar:

- **INITRD_MODULES** es la lista de los módulos cargados por el initrd.
- **MODULES_LOADED_ON_BOOT** es la lista de módulos cargados por init (por lo tanto, después de initrd).

Verá una diferencia entre la lista de los módulos indicada para el initrd y la lista real cargada durante su creación. El comando **mkinitrd** de openSUSE permite especificar funcionalidades preestablecidas durante la creación del **initrd** (opción **-f "feature1 feature2 etc"**). Por ejemplo, la funcionalidad "usb" cargará todos los módulos relacionados con el soporte USB, lo que resulta vital si dispone de un teclado USB o si inicia el sistema desde un dispositivo USB o un disco externo.

d. Debian y Ubuntu: /etc/modules

En Debian y Ubuntu, basta con añadir los nombres de los módulos que se deben cargar en el archivo **/etc/modules**. Después del nombre de los módulos, puede indicar parámetros. Para cada línea se ejecuta un modprobe. Puede añadir un comentario después de una #. Es el archivo**/etc/init.d/modutils** o el archivo de configuración upstart **/etc/init/kmod.conf**(Ubuntu) quien carga los módulos en el momento del inicio mediante init System V, o**/etc/init/module-init-tools.conf** (Debian).

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should
# be loaded at boot time, one per line. Lines beginning with
#  "#" are ignored.
# Parameters can be specified after the module name.

lp
rtc
```

5. Parámetros dinámicos

a. /proc y /sys

/proc y **/sys** son sistemas de archivos virtuales que contienen información relativa al núcleo en curso de ejecución. La versión 2.4 del núcleo sólo conoce **/proc**, donde se agrupa toda la información. La versión 2.6 del núcleo ha modificado la función de **/proc** para delegar una parte de sus tareas a **/sys**.

Al tratarse de sistemas de archivos virtuales, no ocupan espacio ni en la memoria, ni en cualquier disco. Uno no debe dejarse engañar por el tamaño de los seudoarchivos que contiene. ¡No intente suprimir **/proc/kcore** para ganar espacio! Se puede leer y visualizar directamente todos estos archivos (o casi).

Los archivos de **/proc** le proporcionarán mucha información sobre el sistema:

- **interrupts**: los parámetros IRQ.
- **cpuinfo**: detalles sobre sus procesadores.
- **dma**: los parámetros DMA.
- **ioports**: los puertos de memoria de E/S.
- **devices**: los periféricos presentes.
- **meminfo**: el estado global de la memoria.
- **loadavg**: la carga del sistema.
- **uptime**: tiempo transcurrido desde el arranque.
- **version**: detalles de la versión de Linux.
- **modules**: idéntico al resultado de Ismod.
- **swaps**: lista y estado de las particiones de intercambio.
- **partitions**: lista y estado de las particiones conocidas del sistema.
- **mounts**: montaje de los sistemas de archivos.
- **pci**: detalles del bus PCI.

```
$ cat /proc/cpuinfo

processor        : 0
vendor_id        : GenuineIntel
cpu family       : 6
model            : 26
model name       : Intel(R) Xeon(R) CPU           L5520  @ 2.27GHz
stepping         : 5
microcode        : 0xffffffff
cpu MHz          : 2266.689
cache size       : 8192 KB
fpu              : yes
fpu_exception    : yes
cpuid level      : 11
wp               : yes
flags            : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx lm constant_tsc up
rep_good nopl pni ssse3 cx16 sse4_1 sse4_2 popcnt hypervisor lahf_lm
bogomips         : 4533.37
clflush size     : 64
cache_alignment  : 64
address sizes    : 40 bits physical, 48 bits virtual
power management:
$ cat /proc/version
Linux version 3.2.0-4-amd64 (debian-kernel@lists.debian.org) (gcc version 4.6.3
(Debian 4.6.3-14) ) #1 SMP Debian 3.2.63-2+deb7u2
```

/proc contiene subdirectorios que agrupan información por tema.

- **/proc/scsi**: información sobre el bus SCSI.
- **/proc/ide**: información sobre el bus IDE.
- **/proc/net**: información sobre la red.
- **/proc/sys**: parámetros y configuración dinámica del núcleo.
- **/proc/<PID>**: información sobre el proceso PID.

Algunas entradas de los sistemas de archivos **/proc/sys** y **/sys** son diferentes de los demás, ya que se puede modificar su contenido y el núcleo tiene en cuenta las modificaciones directamente, sin que sea preciso iniciar de nuevo la máquina.

Por ejemplo, veamos cómo activar el forwarding IP y pasar el número de manejadores de archivos de 8192 a 16384.

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
# echo "16384" > /proc/sys/fs/file-max
```

He aquí otro ejemplo relacionado con el capítulo anterior y la gestión de la memoria alta y baja. En los antiguos núcleos (por ejemplo 2.6.18) se podía modificar el ratio de protección de la memoria baja lowmem modificando **lower_zone_protection**. Se indica que el sistema debe intentar proteger lo mejor posible una zona de 300 MB de la memoria lowmem.

```
# echo 300>/proc/sys/vm/lower_zone_protection
```

Los núcleos recientes usan **lowmem_reserve_ratio**. Los tres valores se expresan en ratio 1/n de páginas de memoria que se deben proteger por zona de memoria. Para reservar más páginas, basta con reducir los valores:

```
# echo "128 128 32" >/proc/sys/vm/lowmem_reserve_ratio
```

b. sysctl

Los valores modificados en caliente no se registran. En caso de un nuevo inicio, hay que empezar de nuevo. El archivo **rc.sysinit** llama al comando **sysctl**, que actúa sobre estos parámetros. Para que los valores se queden permanentes (nueva instalación a cada inicio), hay que modificar el archivo **/etc/sysctl.conf** o crear/modificar uno de los archivos ubicados en**/etc/sysctl.d/*.conf**. Puede buscar las modificaciones manualmente.

```
# sysctl -e -p /etc/sysctl.conf

# sysctl -a
...
dev.raid.speed_limit_max = 100000
dev.raid.speed_limit_min = 100
net.token-ring.rif_timeout = 60000
net.ipv4.conf.eth1.arp_filter = 0
net.ipv4.conf.eth1.tag = 0
net.ipv4.conf.eth1.log_martians = 0
net.ipv4.conf.eth1.bootp_relay = 0
net.ipv4.conf.eth1.proxy_arp = 0
net.ipv4.conf.eth1.accept_source_route = 1
net.ipv4.conf.eth1.send_redirects = 1
net.ipv4.conf.eth1.rp_filter = 1
net.ipv4.conf.eth1.shared_media = 1
net.ipv4.conf.eth1.secure_redirects = 1
```

. . .

Compilar un núcleo

1. Obtener las fuentes

a. Fuentes oficiales

Las fuentes oficiales del núcleo están disponibles en el sitio kernel.org. Se denominan vanilla. Un núcleo (o kernel) vanilla es un núcleo bruto, sin parches añadidos, procedente directamente de los desarrolladores que colaboran en el núcleo, y no ha sido adaptado aún a ninguna distribución.

Una vez se haga con las fuentes, debe configurar, compilar e instalar un núcleo, y si es preciso, crear un **initrd** antes de poder utilizarlo.

El núcleo se proporciona en forma de carpeta comprimida que debe abrir con las herramientas adecuadas.

El formato actual de compresión empleado es el formato XZ. Probablemente deberá instalar las herramientas xz-utils (Debian, Ubuntu) o xz (Fedora, Red Hat). La opación asociada de tar es "J".

```
# ls
linux-3.16-rc3.tar.xz
# tar tJf linux-3.16-rc3.tar.xz
linux-3.16-rc3/
linux-3.16-rc3/.gitignore
linux-3.16-rc3/.mailmap
linux-3.16-rc3/COPYING
linux-3.16-rc3/CREDITS
linux-3.16-rc3/Documentation/
linux-3.16-rc3/Documentation/.gitignore
linux-3.16-rc3/Documentation/00-INDEX
linux-3.16-rc3/Documentation/ABI/
linux-3.16-rc3/Documentation/ABI/README
linux-3.16-rc3/Documentation/ABI/obsolete/
linux-3.16-rc3/Documentation/ABI/obsolete/proc-sys-vm-nr_pdflush_threads
linux-3.16-rc3/Documentation/ABI/obsolete/sysfs-bus-usb
linux-3.16-rc3/Documentation/ABI/obsolete/sysfs-class-rfkill
linux-3.16-rc3/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-koneplus
linux-3.16-rc3/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-kovaplus
linux-3.16-rc3/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-pyra
linux-3.16-rc3/Documentation/ABI/removed/
...
```

Se deben ubicar las fuentes del núcleo en **/usr/src/Linux** o **/usr/src/kernels**, según la distribución. Si en el sistema de archivos se encuentran varias versiones de los archivos fuente del núcleo, Linux debe ser el vínculo simbólico hacia la fuente del núcleo actual.

```
# ls -l
total 74944
lrwxrwxrwx  1 root src        13 2014-06-29 23:11 linux -> linux-3.16-rc3
drwxrwxr-x 23 root root     4096 2014-06-29 23:11 linux-3.16-rc3
```

b. Fuentes de la distribución

Las distribuciones vienen con núcleos que menudo están parcheados. Estas modificaciones conllevan varios aspectos: drivers añadidos, backports (actualizaciones retroactivas, añadido de funcionalidades procedentes de una versión ulterior) de núcleos más recientes, actualizaciones de seguridad, funcionalidades añadidas, etc. Se aplican a un núcleo vanilla y a menudo en un orden preciso.

Compilar un núcleo no es una nadería. Si ha firmado un contrato de asistencia con el editor de la distribución (por ejemplo, para una versión Servidor de Red Hat), perderá dicha asistencia si compila el núcleo usted mismo. Si tiene problemas con el núcleo por defecto, debería pensar en:

- comprobar si una actualización oficial corrige su problema;
- subir su problema al foro de asistencia del editor;

Si a pesar de todo desea recompilar el núcleo, el trámite es el mismo que para las fuentes oficiales, sólo que debe instalar el paquete de las fuentes, lo que debería tener como efecto la instalación simultánea de todas las herramientas necesarias para la compilación.

El paquete se llama kernel-sources en openSUSE y Red Hat. En cuanto a este último, diríjase al sitio Red Hat, ya que el paquete **kernel-sources** no es necesario para compilar nuevos módulos (hace falta el paquete kernel-devel) y no es muy fácil obtenerlos. En Debian, el paquete se llama**linux-source-3.x.y** (x e y representan la versión del núcleo). Debe instalar estos paquetes con las herramientas adaptadas a su distribución.

2. Las herramientas necesarias

Para compilar el núcleo de Linux, es necesario disponer de determinadas herramientas:

- el compilador C;
- las librerías de desarrollo C estándar;
- la librería ncurses (para menuconfig);
- la librería qt3 (para xconfig);
- las herramientas Make;
- los modutils;
- mkinitrd;
- concentración;
- nervios de acero;
- paciencia.

La distribución proporciona estas herramientas de forma automática, salvo las tres últimas...

3. Configuración

a. El .config

Antes de lanzar la compilación del núcleo, debe seleccionar las opciones, funcionalidades y drivers que desea conservar o no. A continuación, le mostraremos varias formas de hacerlo. Luego se guarda esta configuración en un archivo **`.config`** de la carpeta de fuentes. Este archivo contiene un gran número de variables: cada una de ellas corresponde a una opción del núcleo y pueden adoptar tres valores:

- **y**: la funcionalidad está presente e integrada dentro del núcleo monolítico, o, si depende de un módulo, integrada dentro de ese módulo;
- **m**: se compilará la funcionalidad en forma de módulo;
- **n**: la funcionalidad no está.

En el último caso, este valor está presente pocas veces. Basta con que la opción no esté en el archivo para que no se active durante la compilación. En este caso, se comenta la línea correspondiente simplemente con una # delante.

El ejemplo siguiente muestra que la opción 64BIT (compilación para procesadores de 64 bits) está activa.

```
# grep -i 64bit .config
CONFIG_64BIT=y
```

El ejemplo siguiente muestra las líneas de configuración asociadas a las funcionalidades del sistema de archivos ext4. Se compila el soporte del sistema de archivos ext4 en forma de módulo (primera línea). Las funcionalidades adicionales de ext4 (atributos extendidos, ACL, seguridad, atributos nfs4) dependen de la primera línea. Por lo tanto, estarán en el núcleo y no en uno o varios módulos aparte.

```
# grep -i ext4 .config
CONFIG_EXT4_FS=y
CONFIG_EXT4_USE_FOR_EXT23=y
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_EXT4_FS_SECURITY=y
CONFIG_EXT4_DEBUG=y
```

Algunas opciones dependen de otras. A no ser que sepa con exactitud lo que hace, no debería editar el archivo **.config** manualmente para efectuar modificaciones en él, pues corre el riesgo de romper dependencias durante la compilación. Lo más sencillo consiste en pasar por las etapas de configuración expuestas a continuación.

Evite modificar **.config** manualmente. Utilice mejor las herramientas facilitadas desde las fuentes o por su distribución.

b. Recuperar la configuración del núcleo

Puede acceder a la configuración actual del núcleo desde varios sitios. Si un núcleo (o sus fuentes) proceden de un paquete de la distribución, es probable que el archivo .config esté ya dentro de **/usr/src/linux** o en otro sitio. En este caso, debe consultar la documentación oficial.

En openSUSE, SLES, Red Hat, Debian o Ubuntu por ejemplo, el directorio **/boot** contiene una copia del .config que ha servido para la compilación del núcleo.

```
# ls -1 /boot/config-*
/boot/config-3.11.10-301.fc20.x86_64
/boot/config-3.14.8-200.fc20.x86_64
/boot/config-3.16.0-0.rc2.git3.1.fc21.x86_64
```

En este caso, puede reutilizar esta configuración manualmente. Tóme la versión más cercana del núcleo que va a recompilar:

```
# cp /boot/config-3.16.0-0.rc2.git3.1.fc21.x86_64
/usr/src/kernels/linux/.config
```

A menudo se configuran los núcleos con dos opciones interesantes, desgraciadamente ausentes de muchas distribuciones:

```
# grep -i KCONF .config
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
```

La primera permite colocar el contenido del .config en el propio núcleo durante la compilación. La segunda permite acceder a esta configuración desde el sistema de archivos virtual **/proc**mediante

el archivo **/proc/config.gz**. Se comprime el seudoarchivo en formato gzip. Para leerlo, utilice el comando **zcat**.

```
# zcat /proc/config.gz > /usr/src/linux/.config
```

c. make oldconfig

El método precedente presenta unos inconvenientes en particular cuando se trata de recuperar la configuración de un núcleo más reciente o más antiguo:

- puede que algunas funcionalidades hayan desaparecido;
- y que se hayan añadido otras.

En este caso, partir de un archivo de configuración inadecuado puede tener consecuencias nefastas. Para evitar cualquier problema, lo mejor es utilizar la posibilidad que ofrecen las fuentes del núcleo (o más bien Makefile): recuperar la antigua configuración, analizarla, indicar los cambios y preguntar qué hacer. Esto se hace con el comando **make oldconfig**.

En el ejemplo siguiente, se vuelve a coger la configuración de un núcleo 3.14.8-200-rc2 de origen Fedora para un núcleo vanilla 3.16-rc3. Hemos truncado muchos de los avisos que aparecen, ya que el núcleo vanilla no contiene ciertas opciones procedentes de parches específicos del núcleo de la distribución. A continuación, el nuevo núcleo dispone de opciones que no estaban en origen. Debe contestar a cada una de las preguntas, lo que resulta fastidioso.

```
# make oldconfig
scripts/kconfig/conf --oldconfig Kconfig
.config:1283:warning: symbol value 'm' invalid for CGROUP_NET_PRIO
*
* Restart config...
*
*
* General setup
*
Cross-compiler tool prefix (CROSS_COMPILE) []
Compile also drivers which will not load (COMPILE_TEST) [N/y/?] n
Local version - append to kernel release (LOCALVERSION) []
Automatically append version information to the version string
(LOCALVERSION_AUTO) [N/y/?] n
Kernel compression mode
> 1. Gzip (KERNEL_GZIP)
  2. Bzip2 (KERNEL_BZIP2)
  3. LZMA (KERNEL_LZMA)
  4. XZ (KERNEL_XZ)
  5. LZO (KERNEL_LZO)
  6. LZ4 (KERNEL_LZ4)
choice[1-6?]: 1
Default hostname (DEFAULT_HOSTNAME) [(none)] (none)
Support for paging of anonymous memory (swap) (SWAP) [Y/n/?] y
System V IPC (SYSVIPC) [Y/n/?] y
POSIX Message Queues (POSIX_MQUEUE) [Y/n/?] y
Enable process_vm_readv/writev syscalls (CROSS_MEMORY_ATTACH) [Y/n/?] y
open by fhandle syscalls (FHANDLE) [Y/n/?] y
uselib syscall (USELIB) [Y/n/?] (NEW)
```

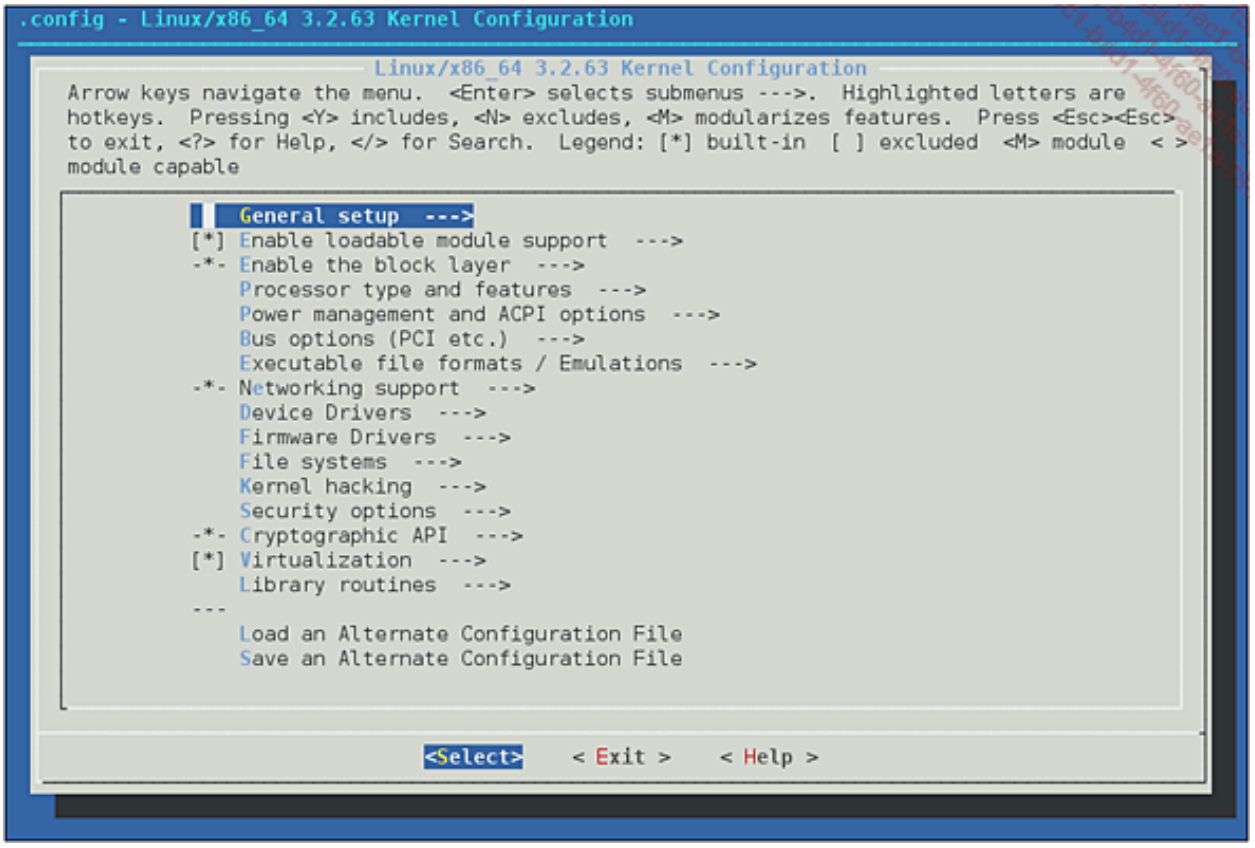
Si pulsa la tecla [Entrar], se activa la opción por defecto. Puede seguir pulsando la tecla [Entrar] hasta el final y luego modificar la configuración con la interfaz textual o gráfica.

d. make menuconfig

El método anterior es práctico cuando se trata de una migración. Sin embargo, quizás querrá utilizar

algo más ameno. Si no dispone de una interfaz gráfica (a menudo ausente en los servidores, ya que es inútil), puede configurar su núcleo pasando por una interfaz en modo consola. A algunos seguramente les recuerde a MS-DOS.

```
# make menuconfig
```



Las opciones de compilación del núcleo empleando menuconfig

Las instrucciones se incluyen en la interfaz (teclee [Entrar], [Y], [N], [M], [?], [/], [Esc]). Observe que si pulsa la tecla [Espacio] se desplegarán todas las opciones posibles. No dude en utilizar la tecla de ayuda para (intentar) entender para qué sirve una opción.

Una vez elegidas las opciones, salga y se generará el archivo .config de manera automática.

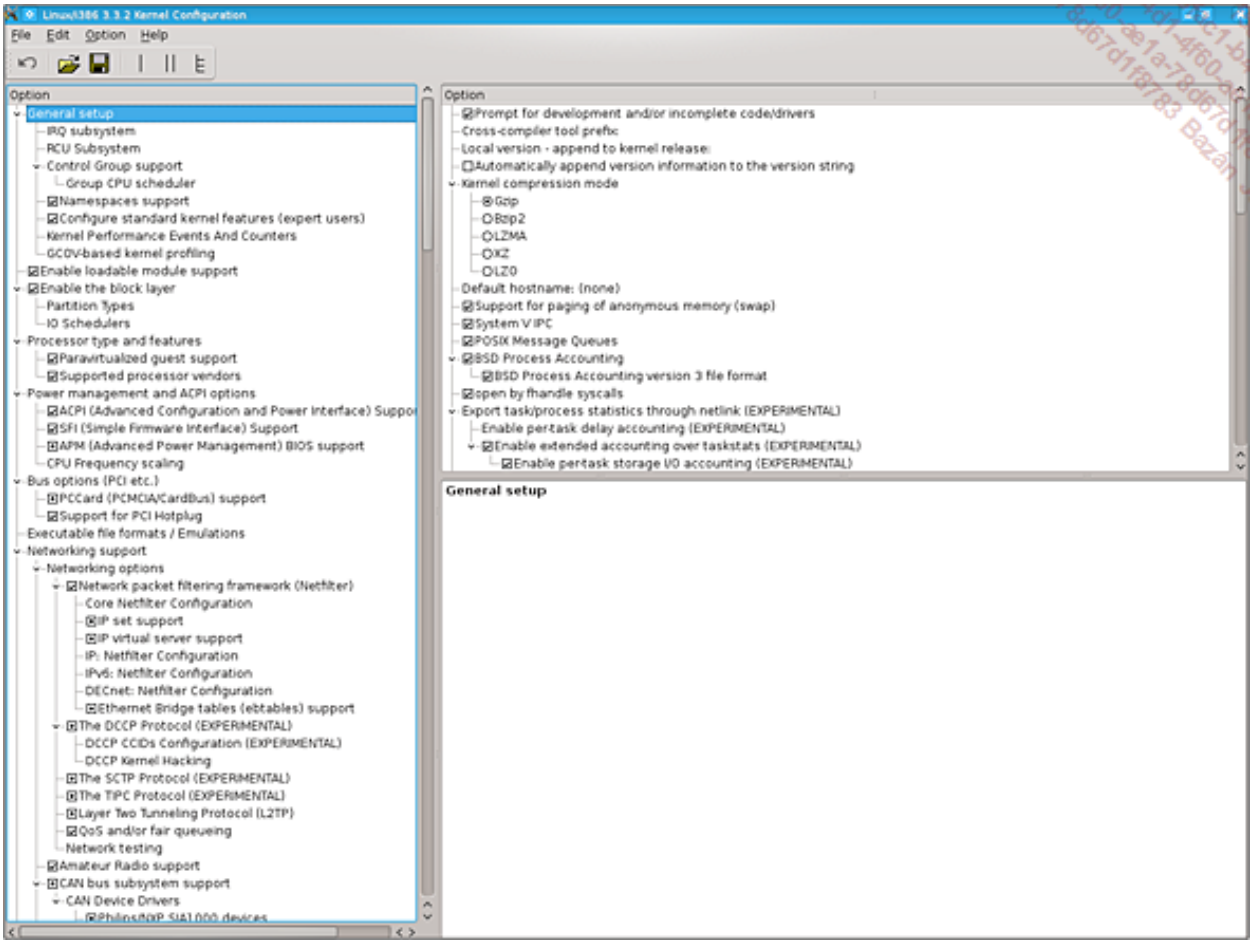
e. make xconfig

Si dispone de un entorno gráfico, he aquí la mejor elección posible. Dispone de una interfaz gráfica para configurar su núcleo. La lista de entradas principales está a la izquierda, en forma de árbol. Elija a la derecha las opciones asociadas. En el cuadro inferior, obtendrá ayuda para cada una de las opciones. Los menús le permiten efectuar una búsqueda entre los valores.

Tendrá que instalar las librerías de desarrollo qt4 (generalmente qt4-devel, o qt-devel) al igual que el compilador c++ (gcc-c++).

Desde la interfaz gráfica, puede cargar una configuración procedente de otro archivo de configuración o exportarla hacia un archivo que elija.

```
# make xconfig
```



Las opciones de compilación empleando xconfig

Si ejecuta make config, la interfaz utilizada será gtk en vez de qt. Será necesario instalar las librerías de desarrollo gtk.

f. Algunas opciones de optimización

Las distribuciones Linux suministran núcleos genéricos. Algunas proponen núcleos específicos para algunos tipos de procesadores (Intel, AMD, etc.). En todos los casos se facilitan versiones de 32 bits y 64 bits. Estas últimas permiten aprovechar los procesadores Intel o AMD recientes (Athlon64, Prescott, Core2, etc.).

Estos núcleos se diseñan para funcionar en un máximo de máquinas, procesadores y dispositivos. Se compilan con opciones estándares y la casi totalidad del soporte de los diferentes periféricos. Así pues, no están optimizados para algunos usos. Le parecerá muy interesante lo que puede aportar la posibilidad de adaptar el núcleo a su máquina:

- Una adecuación perfecta de las optimizaciones del núcleo para su hardware.
- Mayor eficacia según el tipo de uso (servidor, estación de trabajo).
- El soporte de nuevos materiales (núcleo más reciente).
- Una reducción del volumen en disco ocupado por los módulos inútiles.

En cualquier caso, si compila un nuevo núcleo, no desinstale ni borre el antiguo. Consérvelo: en caso de problemas con su núcleo personalizado, tendrá la posibilidad de volver atrás.

Algunas pistas para mejorar su núcleo:

- Active las optimizaciones del núcleo para su procesador. En la entrada **Processor type and features** seleccione, por ejemplo, **Core2 / Newer Xeon** si su máquina dispone de un

procesador Core 2 Duo o superior.

- Modifique el valor **Preemption Model** a **Preemptible kernel** y marque la casilla **Preempt the Big Kernel Clock** para reducir el efecto de latencia del núcleo (para esteblecer dicho modo en el proceso).
- Modifique el valor del **Timer Frequency** a 300 Hz o 1000 Hz para reducir el tiempo de respuesta a los eventos.
- En **CPU Frequency Scaling**, también puede aumentar ligeramente la velocidad de inicio del núcleo pasando el valor **Default CPUfreq Governor** a **performance**. Esto no cambia nada de momento. Puede seguir ajustando después la velocidad del procesador.
- Suprima luego los diferentes módulos que no va a necesitar nunca. Por ejemplo, no hace falta conservar las tarjetas Token Ring o ISDN si no las utiliza jamás.

En cuanto a este último punto, tenga cuidado de no suprimir cualquier cosa. No olvide que, si cambia de máquina, a diferencia de Windows, no tiene que volver a instalar obligatoriamente Linux si el núcleo es bastante genérico (quizá tendrá que bootear en single para modificar los módulos cargados en el inicio). El autor migró de un Core 2 Duo a un Core i5 (con cambio de placa base, tarjeta gráfica, etc.) sin necesidad de reinstalar completamente su sistema. El nuevo sistema puede necesitar módulos desafortunadamente suprimidos.

Piense también en los soportes y periféricos en hotplug o en hotswap. Si por error ha suprimido el driver o una opción del driver, tendrá que recompilar el núcleo o reiniciar a partir del antiguo que haya conservado.

Una buena idea con los núcleos por defecto de las distribuciones consiste en tocar únicamente las opciones de optimización, como las que hemos visto antes, dejando todas las demás opciones por defecto.

¿32 o 64 bits?

Casi todas las máquinas vendidas hoy en día, incluso los ordenadores portátiles, disponen de un conjunto de instrucciones de 64 bits. Hasta la llegada de Windows 7, los ordenadores disponibles para el gran público se vendían con sistemas de 32 bits (Windows XP o Vista, principalmente), a pesar de que existían las versiones de 64 bits. Los 64 bits ofrecen más ventajas que inconvenientes:

- Mejor gestión de la memoria (adiós a la limitación de 4 GB).
- Mayor eficacia en las aplicaciones optimizadas.
- Compatibilidad en el futuro garantiza.
- Adiós al efecto 2000 (consecuencia de pasar a cero todas las fechas y regresar al 13 de diciembre de 1901) ¡Unix en enero de 2038!

A modo de comparación, la transcodificación de un DVD en DVIX con dvdrip (opciones: de una pasada, desenlace inteligente, gran tamaño de redimensionamiento HQ, mp3 192 kbits) se ejecuta a una velocidad de 27 a 30 imágenes por segundo en openSUSE 10.3 en 32 bits (Core 2 Duo e6750). En la misma máquina, la velocidad es de 40-42 imágenes por segundo (en una instalación optimizada para trabajar en 64 bits). O sea, una ganancia media del 25 al 30 %. Se constata lo mismo con un cálculo súper-pi optimizado. La ganancia puede ser real, incluso si no es significativa en un entorno ofimático.

Dado que se puede obtener el núcleo de Linux mediante archivos fuente y que la mayoría de los drivers están incluidos en las distribuciones o en forma de fuentes, cualquier hardware soportado en 32 bits por Linux lo estará también en 64 bits. Se facilitan algunos módulos propietarios (ati, nvidia) en 64 bits. Las distribuciones de 64 bits proporcionan lo necesario (las librerías) para hacer funcionar las aplicaciones de 32 bits de manera nativa.

Si tiene esta posibilidad, pruebe una versión de 64 bits de Linux.

Si vuelve a compilar un núcleo con las extensiones de 64 bits desde una distribución de 32 bits, únicamente el núcleo en 64 bits funcionará. Habría que instalar luego todas las librerías y todas las herramientas en 64 bits, lo que resulta complejo y largo. En este caso, lo mejor es volver a instalar el sistema.

4. Compilación

Ahora que se ha configurado el núcleo, puede iniciar la compilación. Esta etapa es la más larga. El núcleo contiene millones de líneas en lenguaje C y ensamblador. Según las opciones seleccionadas (y, sobre todo, los módulos incluidos), el nivel de ocupación de su máquina y el rendimiento de su procesador, esta etapa requiere desde unos minutos ¡hasta unas horas! En cualquier caso, tiene tiempo para hacer otra cosa.

Inicie el comando make. Si dispone de varios procesadores o de varios cores, agregue el parámetro -j x, donde x es el número de cores: la compilación se hará en paralelo y funcionará mucho más rápido.

```
# make -j 2
scripts/kconfig/conf --silentoldconfig Kconfig
#
# configuration written to .config
#
  SYSTBL  arch/x86/syscalls/../include/generated/asm/syscalls_32.h
  SYSHDR  arch/x86/syscalls/../include/generated/asm/unistd_32_ia32.h
  SYSHDR  arch/x86/syscalls/../include/generated/asm/unistd_64_x32.h
  SYSTBL  arch/x86/syscalls/../include/generated/asm/syscalls_64.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_32.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_64.h
  SYSHDR  arch/x86/syscalls/../include/generated/uapi/asm/unistd_x32.h
  CHK     include/config/kernel.release
  UPD     include/config/kernel.release
  WRAP    arch/x86/include/generated/asm/clkdev.h
  WRAP    arch/x86/include/generated/asm/early_ioremap.h
  WRAP    arch/x86/include/generated/asm/cputime.h
  WRAP    arch/x86/include/generated/asm/mcs_spinlock.h
  CHK     include/generated/uapi/linux/version.h
  UPD     include/generated/uapi/linux/version.h
  CHK     include/generated/utsrelease.h.o
...
  BUILD   arch/x86_64/boot/bzImage
...
Kernel: arch/x86_64/boot/bzImage is ready  (#1)
  Building modules, stage 2.
...
  LD [M]  sound/pci/snd-ens1371.ko
  CC      sound/pci/snd-es1938.mod.o
  LD [M]  sound/pci/snd-es1938.ko
  CC      sound/pci/snd-es1968.mod.o
  LD [M]  sound/pci/snd-es1968.ko
...
  IHEX2FW firmware/keyspan_pda/keyspan_pda.fw
  IHEX2FW firmware/keyspan_pda/xircom_pgs.fw
  IHEX    firmware/cpia2/stv0672_vp4.bin
  IHEX    firmware/yam/1200.bin
  IHEX    firmware/yam/9600.bin
```

La primera columna representa la acción que se realiza en el archivo representado por la segunda columna. CC indica una compilación; LD, la edición de los vínculos; una [M], la acción sobre un módulo, IHEX un archivo binario, IHEX2FW un firmware, etc.

5. Instalación

Si la compilación se efectúa sin error, le quedan dos acciones por hacer. Primero instale los módulos. El comando siguiente los instala en **/lib/module/<version_núcleo>** y crea el archivo de las dependencias asociadas.

```
# make modules_install|more
  INSTALL Documentation/connector/cn_test.ko
  INSTALL Documentation/filesystems/configfs/configfs_example_explicit.ko
  INSTALL Documentation/filesystems/configfs/configfs_example_macros.ko
  INSTALL arch/x86/crypto/blowfish-x86_64.ko
  INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
  INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
  INSTALL arch/x86/crypto/camellia-x86_64.ko
  INSTALL arch/x86/crypto/cast5-avx-x86_64.ko
  INSTALL arch/x86/crypto/cast6-avx-x86_64.ko
  INSTALL arch/x86/crypto/crc32-pclmul.ko
  INSTALL arch/x86/crypto/crc32c-intel.ko
...
  INSTALL /lib/firmware/whiteheat.fw
  INSTALL /lib/firmware/keyspan_pda/keyspan_pda.fw
  INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
  INSTALL /lib/firmware/cpia2/stv0672_vp4.bin
  INSTALL /lib/firmware/yam/1200.bin
  INSTALL /lib/firmware/yam/9600.bin
  DEPMOD  3.16.0-rc3
```

Este segundo comando vuelve a copiar el núcleo y lo necesario asociado en /boot. Según las distribuciones, va a crear también el initrd asociado y posiblemente modificar la configuración del gestor de arranque GRUB o GRUB2.

```
# make install
[root@localhost linux]# make install
sh ./arch/x86/boot/install.sh 3.16.0-rc3 arch/x86/boot/bzImage \
    System.map "/boot"
```

El resultado en /boot es el siguiente:

```
# ll |grep 3.16.0-rc3
-rw-------. 1 root root 26430045  2 jul. 20:20 initramfs-3.16.0-rc3.img
lrwxrwxrwx. 1 root root       27  2 jul. 20:17 System.map -> /boot/System.map-3.16.0-
rc3
-rw-r--r--. 1 root root  3084490  2 jul. 20:17 System.map-3.16.0-rc3
lrwxrwxrwx. 1 root root       24  2 jul. 20:17 vmlinuz -> /boot/vmlinuz-3.16.0-rc3
-rw-r--r--. 1 root root  6210320  2 jul. 20:17 vmlinuz-3.16.0-rc3
```

Observe que no se ha reconstruido el initrd en este caso, lo que no es normal y debería funcionar con la salida de un núcleo estable. Puede reconstruirlo usted mismo, por ejemplo en Debian o Ubuntu, como hemos visto anteriormente:

```
# cd /boot
# mkinitramfs -o initrd.img-3.16.0-rc3 3.16.0-rc3
# update-grub2
```

Respete el orden precedente. Si instala el núcleo antes que sus módulos, no se podrá volver a construir el initrd, ya que aún no se han instalado los módulos que deberían estar presentes. Puede que no se genere el initrd por defecto. En este caso, tendrá que volver a ejecutar este comando después de la instalación con los parámetros correspondientes al núcleo instalado.

Verifique con GRUB (y no GRUB2) si se ha modificado la configuración del gestor de arranque.

```
# grep 3.16.0 /boot/grub/menu.lst
title Fedora 20 - 3.16.0-rc3
 kernel /boot/vmlinuz-3.16.0-rc3 root=/dev/disk/by-id/scsi-
SATA_ST380011A_5JVTH798-part6 vga=0x31a resume=/dev/sda5 splash=silent
showopts
...
```

Si utiliza GRUB2 y el nuevo núcleo no se tiene en cuenta, vuelva a ejecutar grub2-mkconfig:

```
# grub2-mkconfig -o /boot/grub2/grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.16.0-rc3
Found initrd image: /boot/initramfs-3.16.0-rc3.img
Found linux image: /boot/vmlinuz-3.16.0-0.rc2.git4.1.fc21.x86_64
Found initrd image: /boot/initramfs-3.16.0-0.rc2.git4.1.fc21.x86_64.img
Found linux image: /boot/vmlinuz-3.16.0-0.rc2.git3.1.fc21.x86_64
Found initrd image: /boot/initramfs-3.16.0-0.rc2.git3.1.fc21.x86_64.img
Found linux image: /boot/vmlinuz-3.14.8-200.fc20.x86_64
Found initrd image: /boot/initramfs-3.14.8-200.fc20.x86_64.img
Found linux image: /boot/vmlinuz-3.11.10-301.fc20.x86_64
Found initrd image: /boot/initramfs-3.11.10-301.fc20.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-ddfbf80754d642baaaf8188577b23898
Found initrd image: /boot/initramfs-0-rescue-ddfbf80754d642baaaf8188577b23898.img
done
```

No tiene por qué reinstalar GRUB o GRUB2 cuando modifica su archivo de configuración.

Si utiliza LILO, además de modificar **`/etc/lilo.conf`**, debe reinstalarlo con **/sbin/lilo**.

6. Test

Si todas las etapas anteriores han culminado con éxito, sólo le queda encender de nuevo el ordenador y seleccionar el nuevo núcleo en el momento de la carga. Si el boot termina correctamente (acceso a la consola o entorno gráfico), abra una consola y verifique la versión del sistema.

```
$ uname -a
Linux slyserver 3.0.0-rc7 #1 SMP Wed Jul 20 19:20:16
CEST 2011 x86_64 x86_64 x86_64 GNU/Linux
```

7. Otras opciones

Tras la compilación, los archivos intermediarios (archivos objetos) ocupan mucho espacio. Para suprimirlos, utilice el comando siguiente: `# make clean`

Si su distribución cuenta con el sistema rpm de paquetería, puede crear los paquetes del núcleo (fuentes, headers, núcleo) con el comando siguiente: `# make rpm`

Los archivos de periféricos

1. Introducción

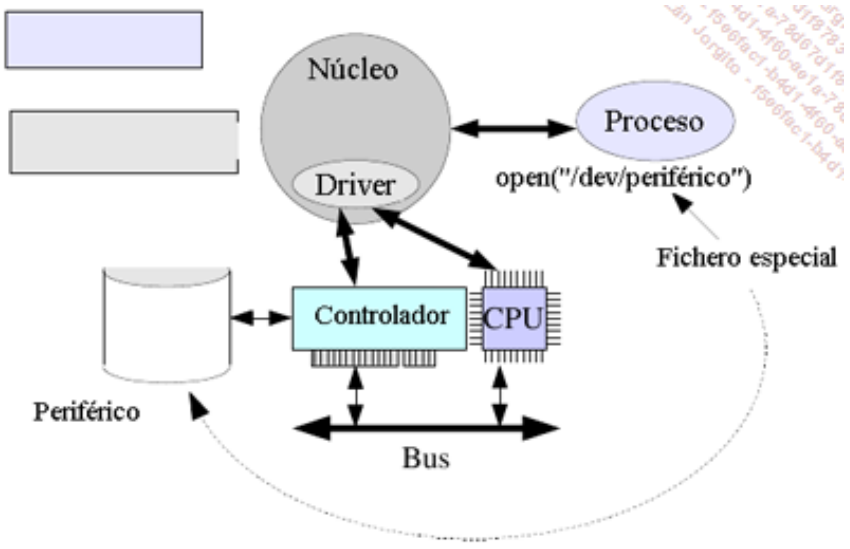
Regresamos al funcionamiento de los periféricos dentro de un ordenador. El principio suele ser el mismo en todos los ordenadores.

Los periféricos están vinculados a un controlador, por ejemplo IDE o SATA para los discos IDE, un controlador SCSI para los discos, lectores y otros escáneres SCSI, o incluso un controlador USB. Un controlador debe saber controlar varios periféricos vinculados a él.

El controlador se comunica con el microprocesador y la memoria con la ayuda del bus (bus de comandos y datos).

En cuanto a Linux, el controlador y sus periféricos se gestionan con la ayuda de drivers (un driver para el controlador, y uno o varios drivers para los periféricos relacionados con él, por ejemplo, un driver para el controlador SCSI, luego un driver para los discos, otro para los escáneres y otro para un CD-Rom). El driver suele ser un módulo complementario del núcleo, entregado por el fabricante o ya presente.

Los periféricos son archivos. En consecuencia, los procesos acceden a los periféricos mediante estos archivos con la ayuda de las funciones en lenguaje C cuyo código está en el núcleo. El proceso debe abrir primero el archivo especial del periférico (primitive open), luego leer (read) o escribir (write) datos desde el periférico o hacia él, como lo haría un archivo normal. Luego el driver del periférico interpreta estas operaciones de lectura/escritura.



Linux accede a los periféricos mediante archivos especiales

En el archivo especial **`/dev/periférico`**, el sistema de gestión de archivos encuentra la información necesaria para dirigirse al controlador correspondiente a través del periférico abierto por un proceso.

2. Archivos especiales

Por convención, se colocan los archivos especiales periféricos en el directorio /dev; éstos disponen de un inodo único, como cualquier otro archivo. Por lo tanto, puede conocer sus atributos mediante el comando **ls -l**.

El primer carácter identifica el tipo de periférico:

- **c**: tipo de periférico en modo carácter;

- **b**: tipo de periférico en modo bloque.

Estos modos diferencian el tipo de intercambio de datos entre el módulo de gestión de archivos y el driver del periférico. En modo carácter, no se utilizan los buffers del sistema y el intercambio se hace byte por byte.

En modo bloque, el sistema accede al periférico mediante un índice que representa las coordenadas del bloque de datos en el soporte. Por lo tanto, es más rápido para periféricos del tipo discos duros.

Los otros dos atributos esenciales de un archivo periférico son el par de información que se encuentra en lugar del tamaño del archivo: el número **mayor** y el número **menor**.

- El número mayor identifica el driver y como consecuencia el controlador de periférico.
- El número menor suele identificar el periférico, pero también puede designar una particularidad del periférico, como la partición de un disco, una ubicación concreta, el número de tarjeta (en caso de que haya varias tarjetas de controladores idénticas, varias tarjetas de sonido, etc.).

Veamos algunos archivos especiales corrientes, según la distribución:

- **/dev/mem**: memoria física.
- **/dev/kmem**: memoria virtual del núcleo.
- **/dev/console**: consola maestra (/dev/syscon).
- **/dev/tty**: entrada/salida estándar del proceso en curso.
- **/dev/mouse**: ratón, a menudo un atajo.
- **/dev/swap**: disco swap primario.
- **/dev/null**: basura UNIX. Se puede escribir en ella. La lectura provoca un EOF.
- **/dev/zero**: retorna cero siempre que se lee. Ideal para llenar archivos vacios para pruebas.
- **/dev/root**: sistema de archivos especial root.
- **/dev/dump**: disco donde el núcleo hace su dump en caso de "panic system".
- **/dev/rmt0**: lector de banda magnética o de cartucho en modo carácter.
- **/dev/fd0**: lector de disquetes en modo bloque.
- **/dev/pts/1**: ídem, pero para Unix SYSTEM V (y Linux).
- **/dev/lp0**: impresora paralela.
- **/dev/ttyS0**: puerto COM1.
- **/dev/ttyS1**: puerto COM2.
- **/dev/psaux**: puerto PS2 para el ratón.
- **/dev/sound**: tarjeta de sonido.
- **/dev/dsp**: controlador DSP de la tarjeta de sonido.
- **/dev/sequencer**: secuenciador MIDI de la tarjeta de sonido.
- **/dev/ide/***: periféricos IDE.
- **/dev/scsi/***: periféricos SCSI.
- **/dev/usb/***: periféricos USB.
- **/dev/hdX**: discos IDE.

- **/dev/sdX**: discos SATA o SCSI.
- etc.

3. Crear un archivo especial

El comando **mknod** permite crear un archivo especial. Aunque Linux disponga de métodos particulares que a menudo se encargan automáticamente de la creación de los archivos periféricos (udev, systemd), en ocasiones la documentación del periférico especifica cómo configurarlo manualmente. En este caso, ¡cuidado! El sistema de archivos **/dev** suele ser de tipo**udev** ¡y totalmente dinámico! Habrá que encontrar otra solución.

```
mknod /dev/periférico tipo mayor menor
```

O bien se integran los drivers en el núcleo durante la compilación de éste, o se compilan estos drivers en forma de módulos complementarios cargados de manera dinámica. Según las distribuciones, el directorio /dev resulta ser a veces un sistema de archivos dinámico (devfs, udev) cuyo contenido cambia en función de la presencia o no de los periféricos. Así, el driver del periférico y un demonio particular **devfsd** o **udevd** se encargan de la creación del archivo periférico. Esto facilita, por ejemplo, la conexión en caliente, como la conexión de periféricos USB bajo demanda: el núcleo detecta el dispositivo, carga el driver correcto y este driver crea dinámicamente el archivo periférico.

A veces, sólo una parte del directorio /dev es dinámico, como el soporte del USB con el sistema de archivos **usbdevfs**.

4. Conocer su hardware

a. Bus PCI

El comando **lspci** proporciona información detallada sobre las tarjetas y los adaptadores relacionados con el bus PCI. Los adaptadores pueden ser los conectados a los puertos de extensión de la placa base pero también los integrados a la placa base (controladores IDE/SATA, tarjetas de red, etc.). Los AGP son considerados bus PCI.

```
# lspci
00:00.0 Host bridge: ATI Technologies Inc RS480 Host Bridge (rev 10)
00:01.0 PCI bridge: ATI Technologies Inc RS480 PCI Bridge
00:11.0 IDE interface: ATI Technologies Inc 437A Serial ATA Controller
00:12.0 IDE interface: ATI Technologies Inc 4379 Serial ATA Controller
00:13.0 USB Controller: ATI Technologies Inc IXP SB400 USB Host Controller
00:13.1 USB Controller: ATI Technologies Inc IXP SB400 USB Host Controller
00:130,2 USB Controller: ATI Technologies Inc IXP SB400 USB2 Host Controller
00:14.0 SMBus: ATI Technologies Inc IXP SB400 SMBus Controller (rev 10)
00:14.1 IDE interface: ATI Technologies Inc Standard Dual Channel PCI IDE
Controller
00:14.3 ISA bridge: ATI Technologies Inc IXP SB400 PCI-ISA Bridge
00:14.4 PCI bridge: ATI Technologies Inc IXP SB400 PCI-PCI Bridge
00:14.5 Multimedia audio controller: ATI Technologies Inc IXP SB400
AC'97 Audio Controller (rev 01)
00:18.0 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
HyperTransport Technology Configuration
00:18.1 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
Address Map
00:18.2 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
DRAM Controller
00:18.3 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
Miscellaneous Control
01:05.0 VGA compatible controller: ATI Technologies Inc RS480 [Radeon
Xpress 200G Series]
```

```
02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8169
Gigabit Ethernet (rev 10)
```

Puede ir aún más lejos en los detalles con la opción **-v** y especificar un adaptador con sus identificadores. Para obtener la información detallada sobre el controlador Ethernet (02:03.0), haga sígueloo siguiente:

```
# lspci -v -s 02:03.0
02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8169
Gigabit Ethernet (rev 10)
        Subsystem: Unknown device 1631:d008
        Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 20
        I/O ports at 8800 [size=256]
        Memory at ff3ffc00 (32-bit, non-prefetchable) [size=256]
        Expansion ROM at ff300000 [disabled] [size=128K]
        Capabilities: [dc] Power Management version 2
```

Cuando es posible, la línea Subsystem indica el fabricante y el modelo exacto del periférico. Se identifica el periférico con un par de valores hexadecimales. El primero es el identificador único del fabricante (vendor). El segundo es el identificador del modelo. Algunos modelos tienen a veces los mismos identificadores, aunque el dispositivo difiera (cambio de chipset). La correspondencia es estática (se inscriben las asociaciones dentro del código compilado). Sirve, entre otras cosas, para cargar los módulos correctos. Si la información no es suficiente, especifique la opción **-vv**.

b. Bus USB

El comando **lsusb** hace lo mismo que lspci, pero para el bus USB:

```
# lsusb
Bus 003 Device 002: ID 04b4:6830 Cypress Semiconductor Corp. USB-2.0 IDE Adapter
Bus 003 Device 001: ID 0000:0000
Bus 008 Device 002: ID 126f:0161 TwinMOS
Bus 008 Device 001: ID 0000:0000
Bus 006 Device 001: ID 0000:0000
Bus 007 Device 001: ID 0000:0000
Bus 005 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 004 Device 003: ID 046d:092e Logitech, Inc.
Bus 004 Device 002: ID 046d:c50e Logitech, Inc. MX-1000 Cordless Mouse Receiver
Bus 004 Device 001: ID 0000:0000
Bus 002 Device 002: ID 045e:00dd Microsoft Corp.
Bus 002 Device 001: ID 0000:0000
```

Cuando es posible, Linux indica cuáles son los nombres de los periféricos mediante una base de identificadores, de la misma manera que para las tarjetas PCI. Estos identificadores también sirven para determinar qué driver USB cargar.

Como en el caso de **lspci**, obtendrá más información con `-v` y con `-d`:

```
# lsusb -d 046d:092e
Bus 004 Device 003: ID 046d:092e Logitech, Inc.
# lsusb -v -d 045e:00dd

Bus 002 Device 002: ID 045e:00dd Microsoft Corp.
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
```

```
  bMaxPacketSize0         8
  idVendor           0x045e Microsoft Corp.
  idProduct          0x00dd
  bcdDevice            1.73
  iManufacturer           1 Microsoft
  iProduct                2 Comfort Curve Keyboard 2000
  iSerial                 0
  bNumConfigurations      1
...
```

c. Recursos físicos

El sistema de archivos virtual /proc está lleno de información sobre su hardware. Veamos una lista no exhaustiva.

Interrupciones

```
# cat /proc/interrupts
           CPU0       CPU1
   0:     692294     559242   IO-APIC-edge      timer
   1:          1          1   IO-APIC-edge      i8042
   8:          1          0   IO-APIC-edge      rtc
   9:          0          0   IO-APIC-fasteoi   acpi
  12:          0          4   IO-APIC-edge      i8042
  16:        591     430773   IO-APIC-fasteoi   ahci, uhci_hcd:usb1,
ohci1394, nvidia
  17:         39      15025   IO-APIC-fasteoi   libata
  18:        285      39829   IO-APIC-fasteoi   ehci_hcd:usb3,
uhci_hcd:usb4, uhci_hcd:usb7
  19:          0          0   IO-APIC-fasteoi   uhci_hcd:usb6
  21:      11107         28   IO-APIC-fasteoi   uhci_hcd:usb2
  22:     397327       3429   IO-APIC-fasteoi   libata, libata, HDA Intel
  23:          1       2533   IO-APIC-fasteoi   uhci_hcd:usb5, ehci_hcd:usb8
4347:         40      24310   PCI-MSI-edge      eth0
 NMI:          0          0
 LOC:    1251342    1251335
 ERR:          0
```

Canales DMA

```
# cat /proc/dma
 4: cascada
```

Intervalos de direcciones de entradas-salidas

```
# cat /proc/ioports
0000-001f: dma1
0020-0021: pic1
0040-0043: timer0
0050-0053: timer1
0060-006f: keyboard
0070-0077: rtc
0080-008f: dma page reg
00a0-00a1: pic2
00c0-00df: dma2
00f0-00ff: fpu
0290-0297: pnp 00:06
  0295-0296: w83627ehf
03c0-03df: vesafb
03f8-03ff: serial
```

```
0400-041f: 0000:00:1f.3
  0400-041f: i801_smbus
0480-04bf: 0000:00:1f.0
0800-087f: 0000:00:1f.0
  0800-0803: ACPI PM1a_EVT_BLK
  0804-0805: ACPI PM1a_CNT_BLK
  0808-080b: ACPI PM_TMR
  0810-0815: ACPI CPU throttle
  0820-082f: ACPI GPE0_BLK
0cf8-0cff: PCI conf1
9400-940f: 0000:00:1f.2
9480-948f: 0000:00:1f.2
  9480-948f: libata
...
```

Periféricos (bloque, carácter)

```
# cat /proc/devices
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 /dev/vc/0
  4 tty
  4 ttyS
...
 14 sound
...
171 ieee1394
180 usb
189 usb_device
195 nvidia
253 rtc
254 usb_endpoint

Block devices:
  7 loop
  8 sd
  9 md
 11 sr
 65 sd
...
253 device-mapper
254 mdp
```

Particiones

```
# cat /proc/partitions
major minor  #blocks  name

   8     0  160836480 sda
   8     1   41945683 sda1
   8     2     530145 sda2
   8     3   41945715 sda3
   8     4          1 sda4
   8     5    2104483 sda5
   8     6   74308626 sda6
   8    16  156290904 sdb
   8    17  156288321 sdb1
   8    32  293036184 sdc
   8    33  293033601 sdc1
   8    48     503808 sdd
```

```
   8     49     503792 sdd1
```

Procesadores

```
# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 15
model name      : Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz
stepping        : 11
cpu MHz         : 1998,000
cache size      : 4.096 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
syscall nx lm constant_tsc pni monitor ds_cpl vmx smx est tm2 ssse3
cx16 xtpr lahf_lm
bogomips        : 6804.61
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:
...
```

d. Otras herramientas

Las distintas distribuciones entregan juegos de herramientas complementarias, como **hwinfo** o**dmidecode** más estándar.

hwinfo

La herramienta **hwinfo** detecta su hardware y le da una lista (de manera corta con la opción `--short`). Este comando realiza un rastreo por todo su hardware, que devuelve la información:

```
# hwinfo --short
cpu:
                       Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz, 2000 MHz
...
keyboard:
  /dev/input/event4    Microsoft Comfort Curve Keyboard 2000
mouse:
  /dev/input/mice      Logitech MX-1000 Cordless Mouse Receiver
monitor:
                       Generic Monitor
graphics card:
                       ASUSTeK GeForce 8600 GT
sound:
                       ASUSTeK 82801I (ICH9 Family) HD Audio Controller
storage:
                       ASUSTeK 82801IB (ICH9) 2 port SATA IDE Controller
...
network:
  eth0                 ASUSTeK L1 Gigabit Ethernet Adapter
...
```

```
disk:
  /dev/sda            HDT722516DLA380
...
partition:
  /dev/sda1           Partition
...
cdrom:
  /dev/sr0            TSSTcorp CD/DVDW SH-S183A
...
usb controller:
                      ASUSTeK 82801I (ICH9 Family) USB UHCI Controller #4
...
BIOS:
                      BIOS
bridge:
                      ASUSTeK 82G33/G31/P35/P31 Express DRAM Controller
...
Port:
                      Intel 82801I (ICH9 Family) PCI Express Port 1
...
memory:
                      Main Memory
firewire controller:
                      ASUSTeK IEEE 1394 Host Controller
unknown:
...
                      Keyboard controller
...
  /dev/ttyS0          16550A
                      Logitech Camera
  /dev/input/event5   Microsoft Comfort Curve Keyboard 2000
```

Para obtener más detalles, suprima la opción **--short** y especifique, si es preciso, qué componente quiere detallar. Por ejemplo, **--cpu** para el procesador, **--memory** para la memoria, etc. La lista está en el manual del comando.

```
# hwinfo --cpu
01: None 00.0: 10103 CPU
 [Created at cpu.301]
 Unique ID: rdCR.j8NaKXDZtZ6
 Hardware Class: cpu
 Arch: X86-64
 Vendor: "GenuineIntel"
 Model: 6.15.11 "Intel(R) Core(TM)2 Duo CPU     E6750  @ 2.66GHz"
 Features:
fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,
clflush,dts,acpi,mmx,fxsr,sse,sse2,ss,ht,tm,syscall,nx,lm,constant_tsc,
pni,monitor,ds_cpl,vmx,smx,est,tm2,ssse3,cx16,xtpr,lahf_lm
 Clock: 2666 MHz
 BogoMips: 6804.61
 Cache: 4096 kb
 Units/Processor: 2
 Config Status: cfg=new, avail=yes, need=no, active=unknown
...
# hwinfo --memory
01: None 00.0: 10102 Main Memory
 [Created at memory.61]
 Unique ID: rdCR.CxwsZFjVASF
 Hardware Class: memory
 Model: "Main Memory"
 Memory Range: 0x00000000-0x7ff7ffff (rw)
 Memory Size: 2 GB
 Config Status: cfg=new, avail=yes, need=no, active=unknown
```

dmidecode

La herramienta **dmidecode** no interroga a los dispositivos directamente, sino que lee e interpreta la tabla **DMI** (*Desktop Management Interface*) del ordenador, a veces llamada también **SMBIOS**(*System Management BIOS*). No sólo proporciona información sobre el estado físico actual de la máquina, sino también sobre sus extensiones posibles (por ejemplo, la velocidad máxima del procesador, la cantidad de memoria posible, etc.). A diferencia de **hwinfo**, que interroga a un componente, por ejemplo la CPU, **dmidecode** lee la información tal como la detecta la BIOS y la placa base. Es rápido, a veces más concreto que **hwinfo**, pero en ocasiones da errores, por lo que se recomienda comprobarlo.

Como la salida es mucho más larga que con **hwinfo**, especifique qué información desea con `-s` o `-t` (consulte el manual):

```
# dmidecode -t processor
# dmidecode 2.9
SMBIOS 2.4 present.

Handle 0x0004, DMI type 4, 35 bytes
Processor Information
        Socket Designation: LGA775
        Type: Central Processor
        Family: Pentium 4
        Manufacturer: Intel
        ID: FB 06 00 00 FF FB EB BF
        Signature: Type 0, Family 6, Model 15, Stepping 11
        Flags:
                FPU (Floating-point unit on-chip)
                VME (Virtual mode extension)
                DE (Debugging extension)
                PSE (Page size extension)
                TSC (Time stamp counter)
                MSR (Model specific registers)
                PAE (Physical address extension)
                MCE (Machine check exception)
                CX8 (CMPXCHG8 instruction supported)
                APIC (On-chip APIC hardware supported)
                SEP (Fast system call)
                MTRR (Memory type range registers)
                PGE (Page global enable)
                MCA (Machine check architecture)
                CMOV (Conditional move instruction supported)
                PAT (Page attribute table)
                PSE-36 (36-bit page size extension)
                CLFSH (CLFLUSH instruction supported)
                DS (Debug store) ACPI (ACPI supported)
                MMX (MMX technology supported)
                FXSR (Fast floating-point save and restore)
                SSE (Streaming SIMD extensions)
                SSE2 (Streaming SIMD extensions 2)
                SS (Self-snoop)
                HTT (Hyper-threading technology)
                TM (Thermal monitor supported)
                PBE (Pending break enabled)
        Version: Intel(R) Core(TM)2 Duo CPU E6750 @ 2.66GHz
        Voltage: 1.3 V
        External Clock: 427 MHz
        Max Speed: 3800 MHz
        Current Speed: 3416 MHz
        Status: Populated, Enabled
        Upgrade: Socket LGA775
        L1 Cache Handle: 0x0005
        L2 Cache Handle: 0x0006
```

```
        L3 Cache Handle: 0x0007
        Serial Number: To Be Filled By O.E.M.
        Asset Tag: To Be Filled By O.E.M.
        Part Number: To Be Filled By O.E.M.
```

Observe que los valores en negrita no son los mismos que los devueltos por los otros comandos. Aquí no hay error: el procesador está ”overclockeado” a 3,4 GHz...

```
# dmidecode -s processor-frequency
3416 MHz
```

5. El soporte del USB y del hotplug

a. Los módulos

El **USB** (*Universal Serial Bus*) es un bus de datos en modo serie y *plug and play*. El principio consiste en que, una vez conectado el adaptador USB, el sistema carga el posible driver correspondiente y el equipo funciona en seguida. En USB 2.0, las tasas de transferencia pueden alcanzar 480 Mbits/s. El USB 3 permite una tasa de transferencia de 4,8 Gbits/s.

Para las versiones USB 1.0 y USB 1.1, se reparten el mercado dos tipos de controladores: **UHCI** y**OHCI**.

- Universal Controller Host Interface: desarrollado por Intel.
- Open Controller Host Interface: los demás.

En USB 2.0, el controlador se llama **EHCI**: *Enhanced Host Controller Interface*.

Por último para USB 3.0 el controlador se llama **XHCI**: *eXtended Host Controller Interface*.

Linux gestiona los tres tipos de controladores.

El módulo básico se llama **usbcore**. Propone el conjunto de las API necesarias para otros módulos:

- **ohci_hcd**: soporte OHCI.
- **uhci_hcd**: soporte UHCI.
- **ehci_hcd**: soporte EHCI.
- **xhci_hcd** : soporte XHCI.
- **usb_storage**: capa de acceso a los soportes masivos: discos externos, pendrives, etc.
- **usbhid**: capa de acceso al soporte de los periféricos **HID** (*Human Interface Device*) de tipo teclados, ratones, joystick, etc.
- **snd-usb-audio**: soporte de tarjetas de sonido USB.
- **usbvideo**: soporte de tarjetas de vídeo y adquisición de USB.
- **irda-usb**: soporte de los puertos de infrarrojo USB.
- **usbnet**: soporte de las tarjetas de red USB.
- etc.

Si estos módulos parecen ausentes, pueden que estén integrados directamente en el núcleo. En este caso, una búsqueda en el archivo de configuración del núcleo o con dmesg le dará la información.

```
grep -i xhci_hcd /boot/config-3.13.0-24-generic
       CONFIG_USB_XHCI_HCD=y
```

b. Carga

Se cargan los módulos USB (y otros relacionados con el hardware) de distintas formas:

- Con **ramdisk initial**: actualmente la mayoría de los teclados y ratones son de tipo USB. Por eso, se suele cargar el soporte básico del USB y el módulo usbhid mediante el initrd (Initial Ramdisk). De ahí que a menudo haya un pequeño desfase entre el momento en el cual GRUB carga el núcleo y el momento en el que se puede utilizar el teclado, entre el final del soporte del USB por la BIOS (paso del núcleo al modo protegido) y el momento en que el núcleo se encarga del soporte USB.
- Con **init**: un servicio encargado de la detección del hardware y de la carga de los drivers carga la lista de los módulos correspondiente al hardware.
- Con **kmod**: carga automática de los módulos del núcleo. Cuando el núcleo detecta la presencia de un nuevo periférico USB, puede cargar el módulo correspondiente. Recuerde que un módulo proporciona los identificadores del hardware del cual se encarga. El archivo**modules.usbmap** provee esta correspondencia. Entonces el núcleo ejecuta **modprobe**.
- Con **udev** o **hotplug**: unas reglas permiten especificar acciones cuando llegan nuevos periféricos, entre las cuales se encuentra, por ejemplo, la carga de módulos complementarios.
- Con **systemd**: systemd y udev están vinculados completamente. Un mensaje del bus udev puede desencadenar una acción de systemd para la creación de un dispositivo y la carga de un módulo.
- Manualmente.

c. hotplug, usbmgr

Antes de la llegada de udev y HAL, se utilizaba principalmente el proyecto **Linux Hotplug Project**. hotplug no gestionaba únicamente el USB, sino cualquier hardware conectado, en caliente o no (coldplug), al PC.

Ya no se utiliza hotplug en ninguna de las distribuciones recientes basadas en el núcleo 2.6. Ha dejado su sitio al tándem udev/HAL.

usbmgr era un sistema de hotplug previsto para la gestión del USB. Ya no se utiliza.

d. udev

udev es un sistema de archivos dinámico que sustituye al antiguo **devfs** de los núcleos 2.4. **udev**gestiona todo el árbol /dev. Va a crear y modificar los archivos periféricos presentes en este directorio.

A partir de 2013 udev es un componente de systemd.

A diferencia de hotplug, que se ejecutaba en modo kernel, se inicia **udev** como un servicio clásico, asociado a un sistema de archivos particular. Se sitúa en el espacio del usuario.

```
# mount | grep udev
udev on /dev type tmpfs (rw,mode=0755)
```

El núcleo gestiona eventos y mensajes. **udev** lee los mensajes emitidos por el núcleo y los interpreta. Dispone de reglas que aplica según el tipo de mensaje. Por ejemplo, veamos tres reglas sencillas:

```
KERNEL=="raw1394*",             GROUP="video"
KERNEL=="dv1394*",              SYMLINK+="dv1394/%n", GROUP="video"
KERNEL=="video1394*",           SYMLINK+="video1394/%n", GROUP="video"
```

- **KERNEL**: nombre del evento del núcleo. Aquí aparece un directorio Firewire.

- **GROUP**: el archivo periférico pertenecerá al grupo indicado.
- **SYMLINK**: udev va a crear un vínculo simbólico del archivo periférico hacia la ubicación indicada. Aquí el %n representa el número en el orden de detección.

Es sólo un ejemplo. La formación LPI no requiere escribir reglas elaboradas, sino entender el mecanismo asociado. Con las reglas anteriores, si conecta una videocámara digital gracias al enchufe DV a su tarjeta Firewire, el núcleo va a generar un evento cuyo nombre empieza por "vídeo1394". Se ejecutará la regla correspondiente:

- Aparición de /dev/video1394.
- Modificación de su grupo en vídeo.
- Creación de un vínculo simbólico entre /dev/video1394 y /dev/video1394/0 (el primero).

Se colocan las reglas en **`/etc/udev/rules.d`**.

```
$ ls -l
total 180
-rw-r--r-- 1 root root   191 sep 21  2007 05-udev-early.rules
-rw-r--r-- 1 root root   366 oct  2  2007 40-alsa.rules
-rw-r--r-- 1 root root  2276 sep 24  2007 40-bluetooth.rules
-rw-r--r-- 1 root root  4937 sep 21  2007 50-udev-default.rules
-rw-r--r-- 1 root root   571 sep 22  2007 51-lirc.rules
-rw-r--r-- 1 root root   399 oct 10  2007 55-hpmud.rules
-rw-r--r-- 1 root root 83444 feb 17 14:22 55-libsane.rules
-rw-r--r-- 1 root root   119 sep 21  2007 56-idedma.rules
-rw-r--r-- 1 root root   119 sep 21  2007 60-cdrom_id.rules
-rw-r--r-- 1 root root  1424 sep 21  2007 60-persistent-input.rules
-rw-r--r-- 1 root root  4347 sep 21  2007 60-persistent-storage.rules
-rw-r--r-- 1 root root   918 sep 21  2007 64-device-mapper.rules
-rw-r--r-- 1 root root   725 sep 21  2007 64-md-raid.rules
-rw-r--r-- 1 root root  1290 sep 22  2007 70-kpartx.rules
-rw-r--r-- 1 root root   611 oct  5  2007 70-persistent-cd.rules
-rw-r--r-- 1 root root   325 oct  5  2007 70-persistent-net.rules
 ...
```

Puede modificar las reglas y crear las suyas propias. En este caso, cree un archivo llamado **99-local.rules** (por ejemplo) y coloque dentro sus reglas.

A modo de ejemplo, las diferentes rutas posibles de acceso a los discos y particiones, presentadas en el capítulo Los discos y el sistema de archivos - Acceder a los sistemas de archivos, proceden de reglas udev. La regla siguiente (que depende de otras reglas en amont, lo que se puede deducir por el test de una variable **udev** DETYPE ya ubicada) importa el resultado de un comando **vol_id**que ya conoce para crear vínculos con UUID y con label.

```
# by-label/by-uuid (filesystem properties)
ENV{DEVTYPE}=="partition", IMPORT{program}="vol_id --export $tempnode"
```

Puede conocer qué periféricos están controlados por **udev** con **udevinfo**, así como el caso de un periférico dado, de la manera siguiente:

```
# udevinfo --query=all -n /dev/sda
P: /block/sda
N: sda
S: disk/by-id/scsi-SATA_ST380011A_5JVTH798
S: disk/by-id/ata-ST380011A_5JVTH798
S: disk/by-path/pci-0000:00:14.1-scsi-0:0:0:0
S: disk/by-id/edd-int13_dev80
E: DEVTYPE=disk
E: ID_VENDOR=ATA
E: ID_MODEL=ST380011A
```

```
E: ID_REVISION=8.01
E: ID_SERIAL=SATA_ST380011A_5JVTH798
E: ID_SERIAL_SHORT=5JVTH798
E: ID_TYPE=disk
E: ID_BUS=scsi
E: ID_ATA_COMPAT=ST380011A_5JVTH798
E: ID_PATH=pci-0000:00:14.1-scsi-0:0:0:0
E: ID_EDD=int13_dev80
```

El comando **udevinfo** ha sido sustituido por **udevadm info** en las últimas versiones de udev:

```
# udevadm info --query=all -n /dev/sdb
P: /devices/pci0000:00/0000:00:1f.2/host1/target1:0:0/1:0:0:0/block/sdb
N: sdb
S: disk/by-id/ata-WDC_WD5000AACS-00G8B1_WD-WCAUK0742110
S: disk/by-id/scsi-SATA_WDC_WD5000AACS-_WD-WCAUK0742110
S: disk/by-path/pci-0000:00:1f.2-scsi-1:0:0:0
S: disk/by-id/wwn-0x50014ee257f93cf8
E: UDEV_LOG=3
E: DEVPATH=/devices/pci0000:00/0000:00:1f.2/host1/target1:0:0
/1:0:0:0/block/sdb
E: MAJOR=8
E: MINOR=16
E: DEVNAME=/dev/sdb
E: DEVTYPE=disk
E: SUBSYSTEM=block
E: ID_ATA=1
E: ID_TYPE=disk
E: ID_BUS=ata
E: ID_MODEL=WDC_WD5000AACS-00G8B1
```

Administración de los usuarios

1. Fundamentos

a. Identificación y autenticación

La **identificación** consiste en saber quién es quién para determinar los permisos de la persona que se conecta. Se identifica un usuario mediante un login.

La **autenticación** consiste en aportar la prueba de quiénes somos mediante, por ejemplo, un secreto compartido entre el usuario y el sistema, y que sólo conocen ellos. Se autentifica al usuario con una contraseña.

b. Los usuarios

Un usuario es la asociación de un nombre de conexión, el login, con un UID y al menos un GID.

- **UID**: User ID.
- **GID**: Group ID.

Los UID y los GID suelen ser únicos. El login es único. Sin embargo, se puede considerar la asociación de varios logins al mismo UID, sabiendo que el sistema trabaja a veces con el login.

El UID identifica el usuario (o la cuenta asociada) a lo largo de su conexión. Se utiliza para el control de sus derechos y de los de los procesos que ha iniciado. Lo que se almacena en la tabla de los inodos, dentro de la tabla de los procesos, son los UID y GID, y no los logins.

El usuario dispone de los atributos básicos siguientes:

- un nombre de inicio de sesión llamado login;
- una contraseña;
- un UID;
- un GID correspondiente a su grupo principal;
- una descripción;
- un directorio de inicio de sesión;
- un comando de inicio de sesión.

Hay otros atributos disponibles mediante la utilización de la seguridad de las contraseñas en el archivo shadow (ver apartado correspondiente).

Se suelen asociar los UID de un valor inferior a 100 a cuentas especiales con derechos extendidos. Así el UID de root, el administrador, es 0. Según las distribuciones, a partir de 100, 500 o 1000, y hasta 65.535 (2^{16}-1) aproximadamente son los UID de los usuarios sin privilegios particulares. Estos parámetros pueden modificarse en **`etc/login.defs`**.

Un login tiene en principio un tamaño de 8 caracteres. En realidad, Linux y otros sistemas aceptan un tamaño mayor, pero con la mayoría de los comandos la visualización de los logins, incluso su gestión, está limitada a 8 caracteres.

Un login acepta la mayoría de caracteres. No debe empezar por una cifra. Es posible modificar la lista de los caracteres autorizados y forzar la longitud y la complejidad mediante los mecanismos de autenticación PAM y el archivo **`/etc/login.defs`**.

c. Los grupos

Cada usuario forma parte de al menos un grupo. Un grupo agrupa usuarios. Como para los logins, el GID del grupo siempre acompaña al usuario para controlar sus privilegios. Un usuario puede formar parte de varios grupos. En este caso, hay que distinguir su grupo primario de los grupos secundarios.

Los grupos son también números (GID). Existen grupos específicos para la gestión de algunas propiedades del sistema y, en particular, el acceso a ciertos periféricos.

Cada vez que un usuario crea un archivo, éste recibe como valor de grupo con privilegios el primario del creador. Si el usuario seb tiene como grupo primario **users**, entonces los archivos creados por seb tendrán como grupo con privilegios **users**.

Un usuario dispone de todos los derechos asociados a sus grupos secundarios. Si seb tiene como grupo secundario **vídeo** y un archivo dispone de los derechos de escritura para este grupo, entonces seb tendrá derecho a modificar su contenido.

El comando **id** permite conocer la información esencial sobre un usuario: uid, gid, grupos secundarios.

```
$ id seb
uid=1000(seb) gid=100(users) grupos=100(users),16(dialout),3(sys),33(video)
```

Seb ha creado un archivo. Su propietario es seb y su grupo es el grupo primario de seb: users.

```
$ touch test
$ ls -l test
-rw-r--r-- 1 seb users 0 abr 10 14:30 test
```

d. Las contraseñas

Las contraseñas permiten autenticar a los usuarios. Deben ser lo bastante complejas como para que no se pueden descubrir fácilmente, pero lo bastante intuitivas como para que se puedan recordar. Las contraseñas están cifradas (MD5, SHA, Blowfish, por ejemplo) y no son directamente legibles bajo su forma cifrada por el usuario para que nadie pueda intentar descifrarlas.

Un usuario debería cambiar regularmente su contraseña, no escribirla nunca en ninguna parte ni llevarla encima. Luego veremos cómo se puede obligar a un usuario a aplicar reglas de creación y duración de contraseñas.

Veamos por ejemplo el resultado encriptado por Blowfish (reconocible por el $2a$ que empieza la cadena) de una contraseña:

```
aher874oP47 vale
$2a$10$CqutecUAlTGSFs2BPnVl..ntI8OEdy5j6gLI/cIKhHP4XZISdlGZO
```

2. Los archivos

a. /etc/passwd

El archivo **/etc/passwd** contiene la lista de los usuarios del sistema local. Cualquier usuario puede leerlo. La información que contiene es pública y útil tanto para el sistema como para los usuarios.

Cada línea representa un usuario y se compone de siete campos.

```
Login:password:UID:GID:comment:homedir:shell
```

- Campo 1: el login o nombre de usuario.
- Campo 2: en las antiguas versiones, la contraseña cifrada. Si hay una x, se coloca la contraseña en **/etc/shadow**. Si es un signo de exclamación, se bloquea la cuenta.

- Campo 3: el User ID.
- Campo 4: el GID, o sea, el grupo principal.
- Campo 5: un comentario o descripción. Es un campo de información.
- Campo 6: el directorio de trabajo, personal, del usuario. Es el directorio al que llega cuando se conecta.
- Campo 7: el shell por defecto del usuario. Pero puede ser cualquier otro comando, incluso un comando que prohíbe la conexión.

b. /etc/group

El archivo **/etc/group** contiene la definición de los grupos de usuarios y en cada uno, la lista de los usuarios de los cuales es el grupo secundario. Cada línea se compone de cuatro campos:

```
Group:password:GID:user1,user2...
```

- Campo 1: el nombre del grupo.
- Campo 2: la contraseña asociada. Veremos la explicación justo a continuación.
- Campo 3: el Group Id.
- Campo 4: la lista de usuarios que forman parte de este grupo.

Es inútil colocar en el cuarto campo a los usuarios que tienen este grupo como grupo principal; lo maneja el sistema.

Le puede sorprender la presencia de un campo de contraseña para los grupos. En la práctica se utiliza pocas veces. Como por supuesto es imposible iniciar sesión como grupo, la explicación está en otra parte. Un usuario tiene derecho a cambiar de grupo para coger, de manera temporal al menos, un grupo secundario como grupo primario con el comando **newgrp**.

En este caso, el administrador puede establecer una contraseña para el grupo para proteger el acceso a este grupo como grupo principal.

c. /etc/shadow

El archivo **/etc/shadow** acompaña al archivo **/etc/passwd**. Ahí se almacenan, entre otras cosas, las contraseñas cifradas de los usuarios. Para ser más precisos, contiene toda la información sobre las contraseñas y su validez en el tiempo. Cada línea se compone de 9 campos separados por ":":

```
bean:$2a$10$AjADxPEfE5iUJcltzYA4wOZO.f2UZ0qP/8EnOFY.P.m10HifS7J8i:13913
:0:99999:7:::
```

- Campo 1: login.
- Campo 2: contraseña cifrada. El xx inicial indica el tipo de cifrado.
- Campo 3: número de días desde el 1º de enero de 1970 hasta el último cambio de contraseña.
- Campo 4: número de días sin poder cambiar la contraseña (0: se puede cambiar en cualquier momento).
- Campo 5: número de días a partir de los cuales se debe cambiar la contraseña.
- Campo 6: número de días antes del vencimiento de la contraseña durante los cuales se debe avisar al usuario.
- Campo 7: número de días después del vencimiento de la contraseña tras los cuales se desactiva la cuenta.

- Campo 8: número de días desde el 1º de enero de 1970 hasta el momento en el cual se desactivó la cuenta.
- Campo 9: reservado.

En el ejemplo de la línea bean, se ha cambiado la contraseña 13913 días después del 01/01/1970. La contraseña se debe cambiar antes de 0 días, pero seguirá siendo válida, ya que el campo siguiente indica que hay que cambiarla al cabo de 99999 días (273 años) y el campo 5 está vacío (sin obligación de cambio de contraseña). La cuenta se desactiva tras 7 días: no hay riesgo de que ocurra...

Los valores actuales para el cifrado de contraseñas son los siguientes:

- **1**: MD5
- **$2a$**: Blowfish
- **$5**: SHA-256
- **$6**: SHA-512
- Otro: DES

Para conocer la fecha en función del 01/01/1970, utilice el comando date como a continuación, añada el número de días deseado:

```
$ date --date "1 jan 1970 +13984 days"
mar abr 15 00:00:00 CEST 2008
```

d. /etc/gshadow

El archivo **/etc/gshadow** es el equivalente del archivo anterior, pero para los grupos. Sin embargo, algunas de las anteriores distribuciones de Linux no lo soportan por defecto. Se colocan las contraseñas de los grupos en el segundo campo de **/etc/group**.

```
test:$6$FBuQQ/oCSnY/PLbx$MD…rtYrADveathy6xO/8ZYfIpAMF72OZKHUGwJMBox/:root:seb
```

El formato de gshadow es el siguiente:

- Campo 1: nombre del grupo.
- Campo 2: contraseña cifrada. El xx inicial indica el tipo de cifrado, SHA-512 en este caso.
- Campo 3: administradores del grupo (los que pueden modificar la contraseña o los miembros del grupo).
- Campo 4 : miembros del grupo: pueden acceder sin contraseña. Es la misma lista que en /etc/group.

3. Gestión de los usuarios

a. Creación

La creación de un usuario podría ser totalmente manual, ya que Linux (y los demás Unix) se apoya en una serie de comandos que "sólo" modifican archivos planos ya existentes y que crean y vuelven a copiar archivos y carpetas en el sitio correcto con los privilegios correctos.

La creación de un usuario consiste en:

- añadir una línea en **/etc/passwd**,

- añadir una línea en **/etc/shadow**,
- añadir una información si es preciso en **/etc/group**,
- crear el directorio personal y actualizar su contenido con **/etc/skel**,
- cambiar los permisos y el propietario del directorio personal,
- cambiar la contraseña (cifrada).

Se puede crear directamente una cuenta editando los archivos con un editor, aunque no es nada aconsejable. Si desea hacerlo de todas maneras, utilice el comando **vipw**, que actualizará las diversas cachés asociadas a la gestión de las cuentas.

El comando **vipw** admite tres argumentos:

- `-p`: edición de **/etc/passwd**.
- `-g`: edición de **/etc/group**.
- `-s`: edición de **/etc/shadow**.

Evidentemente, en la práctica hay que evitar editar estos archivos a mano, ya que hay otros muchos mecanismos creados a partir del uso de comandos específicos para la creación o modificación de cuentas, como por ejemplo, la comprobación de la complejidad de la contraseña.

Cabe destacar que existe una restricción asociada al formato del login. Responde a una norma POSIX: sólo puede empezar por un carácter alfabético, seguido de caracteres portables (letras, cifras, guión, guión bajo, etc.).

Todo esto se puede llevar a cabo con el comando **useradd**. Añade una nueva cuenta y efectúa las principales operaciones:

- creación del usuario y edición de los archivos;
- creación de un grupo privado del usuario (con el mismo nombre que éste);
- creación del directorio personal, edición y modificación de los derechos.

```
useradd <options> login
```

Si no se ha especificado ninguna opción, se recuperarán los valores por defecto dentro del archivo**/etc/defaults/useradd**.

Se aceptan las opciones principales siguientes:

Opción	Función
-m	Crea también el directorio personal. Se incluye a veces por defecto, pero es mejor comprobar que el directorio personal esté presente después de la utilización del comando si no utiliza esta opción.
-u	Especifica el UDI numérico del usuario, para forzarlo. Dicho de otro modo, se calcula el UID según las reglas del archivo login.defs y los UID existentes.
-g	Especifica el grupo primario del usuario, por GID o por su nombre (variable GROUP).
-G	Especifica los grupos adicionales (secundarios, del usuario) separados por comas (variable GROUPS).
-d	Ruta del directorio personal. En general /home/<login>, pero no se puede especificar cualquier ruta (variable HOME/<login>).
-c	Un comentario asociado a la cuenta. Puede ser cualquiera, pero a veces algunos comandos como **finger** lo pueden utilizar. El usuario puede modificar su contenido con el comando **chfn**.

-k	Ruta del directorio que contiene el esqueleto del árbol del directorio del usuario. Suele ser /etc/skell (variable SKEL).
-s	Shell (intérprete de comandos) por defecto del usuario (variable SHELL). El usuario puede cambiarlo mediante el comando **chsh**.
-p	La contraseña del usuario. ¡Cuidado! ¡La contraseña debe estar ya cifrada! A no ser que se vuelva a copiar la contraseña de una cuenta genérica, lo mejor es usar el comando **passwd**.

El comando siguiente crea la cuenta "roberto" con la mayoría de las opciones básicas especificadas. Es sólo un ejemplo, excepto a veces la `-m`, ya que si no se especifica nada, son las opciones por defecto en relación con las especificadas en el archivo **`/etc/defaults/useradd`**.

```
# useradd -m -u 1010 -g users -G video,dialout,lp -s /bin/bash -d
/home/roberto -c "Cuenta de Roberto" roberto
# grep roberto /etc/passwd
robert:x:1010:100:Cuenta de Roberto:/home/roberto:/bin/bash
```

El comando no crea la contraseña. Hay que hacerlo a mano con el comando **passwd**.

```
# passwd roberto
Changing password for roberto.
Nueva contraseña:
Vuelva a introducir la nueva contraseña:
Contraseña cambiada.
```

b. Seguridad de las contraseñas

Cambiar la contraseña

El comando **passwd** permite gestionar las contraseñas, pero también las autorizaciones de inicio de sesión, así como la mayoría de los campos presentes en **`/etc/shadow`**.

Cualquier usuario tiene derecho a cambiar su contraseña en el plazo especificado por el campo 4 de /etc/shadow. La acción por defecto consiste en cambiar la contraseña del usuario actual. Se requiere la antigua contraseña por seguridad (en particular para evitar que una persona mal intencionada modifique su contraseña a sus espaldas). La inserción está enmascarada.

```
$ id
uid=1000(seb) gid=100(users) ...
$ passwd
Changing password for seb.
Antigua contraseña:
Nueva contraseña:
Vuelva a introducir la nueva contraseña:
Contraseña cambiada.
```

Los módulos **PAM** (*Pluggable Authentication Module*) pueden imponer exigencias más o menos estrictas para el cambio de contraseña: en cuanto a la longitud, que no se base en una palabra del diccionario, etc. Veamos lo que ocurre cuando se quiere utilizar pepe (demasiado corto), qwerty (demasiado largo) y María (diccionario):

```
$ passwd
Changing password for seb.
Antigua contraseña:
Nueva contraseña:
Contraseña incorrecta: demasiado corta
Nueva contraseña:
Contraseña incorrecta: demasiado simple
```

```
Nueva contraseña:
Contraseña incorrecta: basada en una palabra del diccionario
passwd: Número máximo de intentos agotado para el servicio
```

El usuario root tiene derecho a modificar las contraseñas de todos los usuarios del sistema, sin que sea preciso que conozca la contraseña anterior. Aún mejor: puede forzar el uso de una contraseña incluso aunque no haya sido validada por PAM:

```
# passwd seb
Changing password for seb.
Nueva contraseña:
Contraseña incorrecta: basada en una palabra del diccionario
Vuelva a introducir la nueva contraseña:
Contraseña cambiada.
```

Gestionar la validez

Se pueden modificar todos los campos de **/etc/shadow** con el comando **passwd**. Veamos algunas de las opciones disponibles.

Opción	Función
-l	Lock: bloquea una cuenta al añadir un ! delante de la contraseña cifrada.
-u	Unlock: desbloquea la cuenta. No se puede activar una cuenta que no tenga contraseña. Hay que utilizar -f para ello.
-d	(root) Suprime la contraseña de la cuenta.
-n <j>	(root) Duración de vida mínima en días de la contraseña.
-x <j>	(root) Duración de vida máxima en días de la contraseña.
-w <j>	(root) Número de días antes de un aviso.
-i <j>	(root) Período de gracia antes de la desactivación si ha vencido la contraseña.
-S	(root) Estatus de la cuenta.

En el ejemplo siguiente, se ha modificado la cuenta bean de esta manera:

- Debe esperar 5 días tras insertar una nueva contraseña para poder cambiarla.
- Su contraseña es válida 45 días.
- Se le avisa 7 días antes de que deba cambiar la contraseña.
- Si no cambia la contraseña tras 45 días, dispone aún de 5 días antes de que sea bloqueada.

```
# passwd -n 5 -x 45 -w 7 -i 5 bean
Password expiry information changed.
```

Veamos la línea de **/etc/shadow** asociada.

```
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2
:13984:5:45:7:5::
```

El comando **chage** permite hacer más o menos lo mismo. Es un comando del root. Iniciado sin otro argumento que el login del usuario, es interactivo. Observe al final la posibilidad de modificar la fecha del último cambio de la contraseña y una fecha fija de expiración de la contraseña (campo 8):

```
# chage bean
Changing aging information for bean.
        Minimum Password Age [7]:
        Maximum Password Age [40]:
        Password Expiration Warning [10]:
        Password Inactive [5]:
        Last Password Change (YYYY-MM-DD) [2008-04-10]:
        Account Expiration Date (YYYY-MM-DD) [1969-12-31]: 2010-01-01
Aging information changed.
```

Veamos la línea **/etc/shadow** resultante:

```
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2
:13979:7:40:10:5:14610:
```

Se aceptan los parámetros siguientes:

Opción	Función
-m	Mindays: equivale a passwd -n.
-M	Maxdays: equivale a passwd -x.
-d	Fecha de última modificación de la contraseña (desde el 01/01/1970).
-E	Fecha de vencimiento de la contraseña (desde el 01/01/1970).
-I	Inactive: equivale a passwd -i.
-W	Warndays: equivale a passwd -w.
-l	List: muestra todos los detalles.

Los detalles son mucho más legibles con **chage** que con **passwd**:

```
# passwd -S bean
bean PS 04/10/2008 7 40 10 5

# chage -l bean
Minimum:        7
Maximum:        40
Warning:        10
Inactive:       5
Last Change:            abr 10, 2008
Password Expires:       may 20, 2008
Password Inactive:      may 25, 2008
Account Expires:        ene 01, 2010
```

Un usuario cualquiera puede visualizar sus propios detalles, pero se le pedirá su contraseña.

c. Modificación

Utilice el comando **usermod** para modificar una cuenta. Utiliza la misma sintaxis y las mismas opciones que useradd, pero dispone también de una sintaxis complementaria que necesita algunas precisiones.

Opción	Función
-L	Bloquear la cuenta, como passwd -l.
-U	Desbloquar la cuenta, como passwd -u.
-f	Vencimiento: la contraseña expira n días después del 01/01/1970

-e <n>	Vencimiento: la contraseña expira n días después del 01/01/1970.
-u <UID>	Modificar el UID asociado al login. Se modifica en consecuencia el propietario de los archivos que pertenecen al antiguo UID dentro del directorio personal.
-l <login>	Modificar el nombre de login.
-m	Move: implica la presencia de -d para especificar un nuevo directorio personal. Se mueve el contenido del antiguo directorio al nuevo.

d. Eliminación

Suprime un usuario con el comando **userdel**. Por defecto no se suprime el directorio personal. Para ello, debe pasar la opción -r.

```
# userdel -r bean
```

El comando no comprueba la totalidad del sistema de archivos, los archivos que estén fuera de su directorio personal no se borrarán. Es el administrador quien debe buscarlos y borrarlos. Para ello, puede usar el comando **find**.

4. Gestión de los grupos

a. Creación

Puede crear un grupo directamente en el archivo **/etc/group** o bien utilizar los comandos asociados. Si edita el archivo manualmente, utilice el comando **vigr** (o vipw -g).

El comando **groupadd** permite crear un grupo. Su sintaxis sencilla acepta el argumento `-g` para especificar un GID preciso.

```
# grep amigos /etc/group
amigos:!:1234:
```

b. Modificación

El comando **groupmod** permite modificar un grupo. Sus parámetros son los siguientes:

Opción	Función
-n <nombre>	Renombra el grupo.
-g <GID>	Modifica el GID. Cuidado: no se modifica el grupo al que pertenecen los archivos correspondientes.
-A <user>	Añadir el usuario especificado en el grupo (grupo secundario).
-R <user>	Suprimir el usuario especificado del grupo.

```
# groupmod -R seb amigos
# grep amigos /etc/group
amigos:!:1234:
```

c. Eliminación

El comando **groupdel** permite suprimir un grupo. Primero el comando comprueba si el grupo que desea suprimir es el grupo primario de un usuario. En este caso, no se permite suprimir el grupo.

Pero si finalmente se puede, no se efectúa más acción que la de suprimir la línea correspondiente en /etc/group. Le corresponde a usted comprobar el sistema de archivos (y la configuración de las aplicaciones si es necesario) para suprimir cualquier traza de este grupo.

```
# groupdel amigos
```

d. Contraseña

El comando **gpasswd** permite asignar una contraseña a un grupo. Esta contraseña se almacena en /etc/group, o, en el caso más frecuente, en /etc/gshadow. Modifique la contraseña del grupo test según este modelo:

```
gpasswd test
```

Agregue un usuario al grupo:

```
@ gpasswd -a seb test
```

Defina la lista de todos los miembros del grupo:

```
# gpasswd -M seb, steph test
```

Defina la lista de administradores del grupo, los que podrán gestionar los usuarios y sus contraseñas:

```
# gpasswd -A admin test
```

Si proporciona una contraseña a un grupo, todos los usuarios que conozcan esta contraseña, incluidos los que no forman parte del grupo, podrán usarla. El grupo se convierte en su grupo principal.

```
$ id -gn
uid=1001(steph) gid=1002(steph) grupos=1002(steph)
$ newgrp test
Contraseña:
$ id
uid=1001(steph) gid=1001(test) grupos=1002(steph),1001(test)
```

El acceso al grupo puede restringirse limitando a sus miembros con el parámetro -R. Es decir, los miembros del grupo no tendrán que introducir la contraseña.

5. Comandos adicionales

a. Conversión de los archivos

Seguramente empleará pocas veces los comandos siguientes. Son útiles sobre todo en el contexto de migraciones de servidores Linux hacia otros Unix, y viceversa. Algunos sistemas Unix no utilizan por defecto (hay que activarla después) la gestión de las cuentas con los archivos **shadow**. En este caso, puede ser necesario convertir los archivos **/etc/shadow** y **/etc/passwd** en uno único, **/etc/passwd**. Es la función del comando **pwunconv**.

En el ejemplo siguiente, se ha convertido el archivo **/etc/passwd**. Una vez ejecutado el comando, cualquier traza de **/etc/shadow** ha desaparecido.

```
# pwunconv
# grep bean /etc/passwd
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:1001:100:
toto:/home/bean:/bin/bash
# ls -l /etc/shadow
ls: no puede acceder a /etc/shadow: Ningún archivo o directorio de este tipo
```

Cuidado: el comando **pwunconv** es destructivo. Toda información de la validez de las contraseñas será destruida. Ya no es posible utilizar las diversas opciones de chage o de passwd relativas a los periodos de validez.

El comando **pwconv** hace lo contrario: crea el archivo **/etc/shadow** asociado a **/etc/passwd**, le mueve dentro las contraseñas y pone los ajustes por defecto, tal y como están definidos en el archivo **/etc/login.defs**.

```
# grep bean /etc/passwd
bean:x:1001:100:pepito:/home/bean:/bin/bash
p64p17bicb3:/home/seb # grep bean /etc/shadow
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2
:13984:0:99999:7:::0
```

Los comandos **grpconv** y **grpunconv** hacen lo mismo para los grupos.

b. Verificar la coherencia

Puede ser útil iniciar herramientas de comprobación de la coherencia de los archivos de los grupos y de las contraseñas. Si está acostumbrado a modificar estos archivos manualmente, nada le asegura que todo esté en orden: puede que falte un grupo, que no exista un shell, que un directorio personal esté ausente, etc. El comando **pwck** efectúa una verificación de los archivos **/etc/passwd** y **/etc/shadow** y reporta los errores.

Veamos un ejemplo en el cual se han alterado los datos del usuario bean de manera voluntaria para probar el comando: el grupo no existe, tampoco el shell. Entre tanto, se ha descubierto otro problema en la máquina de prueba.

```
# pwck
Checking `/etc/passwd'
User `suse-ncc': directory `/var/lib/YaST2/suse-ncc-fakehome' does
not exist.
User `bean': unknown group `14400'
User `bean': shell `/bin/bashr' is not executable.
Checking `/etc/shadow'.
```

El comando **grpck** hace lo mismo para los grupos. En este caso, los controles están menos extendidos y se limitan a los duplicados y a la existencia de los usuarios para los grupos secundarios.

```
# grpck
Checking `/etc/group'
Group `users': unknown user `zorg'
```

c. Comprobar las conexiones

Puede trazar las conexiones en su máquina usando dos comandos. El comando **lastlog** se basa en el contenido de /var/log/lastlog. Acepta los parámetros `-u` (especificación de un usuario) y `-t` para buscar las conexiones de los últimos n días.

```
$ lastlog -u seb
Username                 Port     Latest
seb                      pts/4    jue abr 10 15:13:46 +0200 2008
```

El comando **last** hace más o menos lo mismo, pero se basa en **/var/log/wtmp**, que brinda información adicional, como el origen de la conexión (IP, nombre de la consola, etc.) y las fechas de conexión y desconexión, así como la duración de conexión y si el usuario sigue conectado.

```
$ last
seb      pts/1                        Tue Apr 15 14:39   still logged in
seb      pts/4                        Tue Apr 15 12:06   still logged in
seb      pts/6                        Tue Apr 15 10:07   still logged in
seb      pts/3                        Mon Apr 14 13:36 - 15:18 (1+01:42)
seb      pts/1                        Thu Apr 10 15:39 - 11:58 (4+20:18)
seb      pts/4      localhost         Thu Apr 10 15:13 - 15:39  (00:25)
seb      pts/4      localhost         Thu Apr 10 15:13 - 15:13  (00:00)
seb      pts/4      localhost         Thu Apr 10 15:12 - 15:13  (00:00)
...
```

d. Acciones de usuario

El usuario dispone de algunas acciones sobre las informaciones de su cuenta. Entre otras cosas, puede:

- cambiar su shell de conexión,
- cambiar sus datos personales,
- cambiar de grupo principal,
- tomar la identidad de otra persona.

Cambiar de shell

El comando **chsh** permite al usuario modificar de manera definitiva (o hasta el próximo comando**chsh**) el shell de conexión. No puede elegir cualquier cosa. El shell (o cualquier otro comando) debe estar en **/etc/shells**. A esta lista se accede mediante el parámetro `-l` del comando. La modificación se hace dentro de **/etc/passwd**. Sólo root tiene derecho a modificarla para otros usuarios. Se especifica el nuevo shell con el parámetro `-s`.

```
$ chsh -l
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
...
$ id
uid=1004(bean) gid=100(users) ...
$ chsh -s /bin/ksh
Changing login shell for bean.
Contraseña:
Shell changed.
# grep bean /etc/passwd
bean:x:1004:100:pepito:/home/bean:/bin/ksh
```

Cambiar el comentario

El usuario puede modificar el comentario del archivo **/etc/passwd** usando el comando **chfn**. Es preferible utilizarlo de manera interactiva (se reserva al root el paso de parámetro en modo no interactivo).

```
$ chfn
Changing finger information for bean.
Contraseña:
```

```
Enter the new value, or press ENTER for the default
        Full Name: pepito
        Room Number []: Mister Bean
        Work Phone []: 0102030405
        Home Phone []: 0605040302
        Other:
Finger information changed.
$ grep bean /etc/passwd
bean:x:1004:100:Mister Bean,0102030405,0605040302:/home/bean:/bin/bash
```

Cambiar de grupo primario

El comando **newgrp** permite cambiar de grupo primario de manera temporal, con la condición de que el nuevo grupo especificado sea un grupo secundario del usuario o que el usuario disponga de la contraseña del grupo. Utilizado solo, **newgrp** vuelve al grupo de origen. Las modificaciones son temporales, no se modifica el archivo de las contraseñas.

```
$ id
uid=1004(bean) gid=100(users) grupos=16(dialout),33(video),100(users)
$ newgrp video
$ id
uid=1004(bean) gid=33(video) grupos=16(dialout),33(video),100(users)
```

¿Qué ocurre si intenta coger como grupo primario un grupo que no pertenece a sus grupos secundarios y que está protegido por una contraseña? En el ejemplo siguiente, bean intenta coger como grupo primario grptest.

```
$ id
uid=1004(bean) gid=100(users) grupos=16(dialout),33(video),100(users)
$ newgrp grptest
Password:
$ id
uid=1004(bean) gid=1000(grptest) grupos=16(dialout),33(video),
100(users),1000(grptest)
```

Cambiar de identidad

El usuario puede adoptar la identidad de otra persona mientras dura un comando o toda una sesión. Se suele tratar de root, ya que, como sabe, uno nunca debe conectarse de manera permanente como root (o al menos hay que evitarlo). Por lo tanto, para las tareas administrativas, hay que convertirse en root (u otro usuario) el tiempo necesario.

El comando **su** (*substitute user*) permite abrir una sesión, o ejecutar un shell, o un comando dado, con otra identidad. Obviamente, debe conocer la contraseña de este usuario.

```
su [-c comando] [-s shell] [-] [usuario]
```

Si no se especifica ningún usuario, se utiliza root.

```
$ id
uid=1000(seb) gid=100(users) ...
$ su
Contraseña:
# id
uid=0(root) gid=0(root) grupos=0(root)
```

Observe que se utiliza por defecto el entorno del usuario de origen. No se carga el entorno de root (o de cualquier otro usuario).

Para cargar el entorno completo del usuario de destino, añada el guión como parámetro:

```
# echo $LOGNAME $USER
seb seb
```

```
$ su - bean
Contraseña:
$ echo $USER $LOGNAME
bean bean
```

Para ejecutar de manera puntual un comando con otra identidad, utilice `-c`.

```
$ su -c "make install"
```

El comando **sg** (*substitute group*) es idéntico a **su**, pero emplea un nombre de grupo como argumento.

6. Configuración avanzada

Las descripciones de los archivos siguientes dependen en gran medida de las versiones de los comandos, de sus opciones de compilación y de la política de seguridad aplicada por parte de los editores de las diversas distribuciones. Es posible que algunos archivos no estén presentes y que otros contengan parámetros diferentes.

a. /etc/default/useradd

El archivo **/etc/default/useradd** contiene un cierto número de variables que definen las reglas por defecto que se deben aplicar en el momento de crear un usuario:

- su grupo,
- la raíz de su directorio personal (dónde se ubicará éste),
- si está activo o no,
- el shell,
- su grupo o sus grupos secundarios,
- el lugar donde se sitúa el esqueleto de las cuentas (estructura básica de un directorio de usuario),
- la creación o no de un spool (bandeja) de correo,
- etc.

```
$ cat /etc/defaults/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
GROUPS=video,dialout
CREATE_MAIL_SPOOL=no
```

En vez de editar este comando a mano, puede utilizar los parámetros **--show-defaults** para ver el contenido y **--save-defaults** para modificarlo.

b. /etc/default/passwd

El archivo **/etc/default/passwd** contiene algunas reglas utilizadas por el comando **passwd**para el cifrado de las contraseñas. Es posible definir reglas de cifrado globales, pero también por tipo de archivo, y pasar algunas opciones según el método.

```
$ cat /etc/default/passwd
# Encriptación por defecto
CRYPT=md5

# Encriptación para los archivos (/etc/shadow)
CRYPT_FILES=blowfish

# opción para blowfish
BLOWFISH_CRYPT_FILES=10

# Para NIS
CRYPT_YP=des
```

c. /etc/default/su

El archivo **/etc/default/su** permite configurar el funcionamiento del comando **su**. Por defecto,**su** con el parámetro - instala un nuevo PATH, ya que carga el entorno del usuario de destino. Puede modificarlo e instalar su propio PATH, o conservar el antiguo.

```
# Cambia el PATH incluso sin el guión
ALWAYS_SET_PATH=no

# Path por defecto
PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin

# Path por defecto para root
SUPATH=/usr/sbin:/bin:/usr/bin:/sbin:/usr/X11R6/bin
```

d. /etc/login.defs

Muchos comandos, como, por ejemplo, **login**, **useradd**, **groupadd**, **passwd**, utilizan el archivo**/etc/login.defs** para definir algunos valores por defecto y la validez de los logins. Su contenido puede variar en función de las distribuciones. Suele contener:

- una regla de validez de las cuentas (caracteres autorizados, longitud, etc.);
- los UID mínimo y máximo durante la creación de un usuario;
- los GID mínimo y máximo durante la creación de un grupo;
- los comandos que hay que llamar para crear/modificar/eliminar un usuario;
- las reglas por defecto para la validez de las contraseñas;
- la creación o no de un directorio personal;
- etc.

El contenido del archivo login.defs puede entrar en conflicto con algunas opciones de los demás archivos presentes en /etc/default. En este caso, debe comprobar en los manuales (o manpages) de su distribución qué archivo prevalece.

```
# Login autorizado incluso si el acceso al home es imposible
DEFAULT_HOME        yes

# Path por defecto para el comando login
```

```
ENV_PATH             /usr/local/bin:/usr/bin:/bin

# Ídem para la conexión root
ENV_ROOTPATH         /sbin:/bin:/usr/sbin:/usr/bin

# Plazo en segundos entre dos intentos de login
FAIL_DELAY        3

# Los usuarios en el archivo no ven los mensajes de cnx
HUSHLOGIN_FILE    /etc/hushlogins

# Visualiza la fecha de última conexión
LASTLOG_ENAB        yes

# Se trazan los intentos de conexión de los logins inexistentes
LOG_UNKFAIL_ENAB    no

# tres intentos de login en caso de contraseña incorrecta
LOGIN_RETRIES        3

# Timeout de 60 segundos
LOGIN_TIMEOUT        60

# Ubicación del motd (o de las , separadas por :)
MOTD_FILE    /etc/motd

# tipo de terminales por defecto
TTYTYPE_FILE    /etc/ttytype

# Permiso por defecto de los terminales
TTYGROUP    tty
TTYPERM        0620

# Requiere o no una contraseña para chsh y chfn
CHFN_AUTH        yes

# Restricción del chfn (f:nombre, r:oficina, w:tel trabajo, h:tel casa)
CHFN_RESTRICT        rwh

# PASS_MAX_DAYS    Duración max de la contraseña
# PASS_MIN_DAYS    Plazo mín entre dos cambios de contraseña
# PASS_WARN_AGE    Aviso antes del cambio
PASS_MAX_DAYS      99999
PASS_MIN_DAYS      0
PASS_WARN_AGE      7

# UID/GID Mín y Máx por defecto y sistema para useradd
SYSTEM_UID_MIN          100
SYSTEM_UID_MAX          499
UID_MIN                1000
UID_MAX               60000

#
# Min/max values for automatic gid selection in groupadd
#
# SYSTEM_GID_MIN to SYSTEM_GID_MAX inclusive is the range for
# GIDs for dynamically allocated administrative and system groups.
# GID_MIN to GID_MAX inclusive is the range of GIDs of dynamically
# allocated groups.
#
SYSTEM_GID_MIN          100
SYSTEM_GID_MAX          499
GID_MIN                1000
GID_MAX               60000

# Regexp para los nombres de logins autorizados mediante login/useradd
```

```
CHARACTER_CLASS        [A-Za-z_][A-Za-z0-9_.-]*[A-Za-z0-9_.$-]\?

# Umask por defecto para la creación del homedir
UMASK            022

# Comando ejecutado realmente durante la creación de un grupo
GROUPADD_CMD             /usr/sbin/groupadd.local

# Comando ejecutado realmente durante la creación de un usuario
USERADD_CMD        /usr/sbin/useradd.local

# Comando ejecutado antes de la supresión de un grupo
USERDEL_PRECMD        /usr/sbin/userdel-pre.local

# Comando ejecutado después de la supresión de un grupo
USERDEL_POSTCMD        /usr/sbin/userdel-post.local
```

Además de los comandos **useradd** y **userdel**, le ofrecemos una lista no exhaustiva de los comandos que utilizan los parámetros de este archivo:

- **login**: DEFAULT_HOME, ENV_PATH, ENV_ROOTPATH, FAIL_DELAY, HUSHLOGIN_FILE, LASTLOG_ENAB, LOG_UNKFAIL_ENAB, LOGIN_RETRIES, LOGIN_TIMEOUT, MOTD_FILE, TTYPERM, TTYTYPE_FILE;
- **newusers**: PASS_MAX_DAYS, PASS_MIN_DAYS, PASS_WARN_AGE, UMASK;
- **passwd**: OBSCURE_CHECKS_ENAB, PASS_MAX_LEN, PASS_MIN_LEN, PASS_ALWAYS_ WARN, CRACKLIB_DICTPATH, PASS_CHANGE_TRIES;
- **pwconv**: PASS_MAX_DAYS, PASS_MIN_DAYS, PASS_WARN_AGE.

7. Notificaciones al usuario

a. /etc/issue

Cuando un usuario se conecta por consola, se suele visualizar un mensaje justo antes de la línea de comandos de inserción de su login. El archivo **/etc/issue** contiene este mensaje. Se trata de un mensaje de bienvenida y por este motivo puede contener todo lo que desee. Por defecto, suele contener el nombre de la distribución Linux y el número de versión del núcleo. Admite secuencias de caracteres para poderle aplicar sustituciones.

Veamos el ejemplo del contenido de **/etc/issue** en una distribución openSUSE:

```
$ cat issue
Welcome to openSUSE 11.4 "Celadon" - Kernel \r (\l).
```

b. /etc/issue.net

El mensaje de bienvenida puede ser diferente cuando un usuario se conecta desde una consola remota (telnet, ssh, etc.). A menudo es el mismo, pero sin los caracteres de control relacionados con un shell dado. Para modificar este mensaje específico, edite el contenido del archivo**/etc/issue.net**.

c. /etc/motd

Motd significa *Message of the day*, el mensaje del día. Un vez conectado el usuario desde una consola (local o remota), puede aparecer un mensaje que el administrador puede modificar si edita el archivo **/etc/motd**. Por defecto está vacío. Se puede modificar para, por ejemplo, avisar a los usuarios de que un reboot de mantenimiento tendrá lugar tal día a tal hora, y eso evita mandar n mails...

8. El entorno de usuario

a. /etc/skel

En el momento de crear un usuario y su directorio personal, se establece el entorno de usuario. El entorno contiene, por ejemplo, las variables de entorno, los alias, la ejecución de varios scripts. Se encuentra en archivos cargados en el arranque del intérprete de comandos (shell).

Durante la creación de una cuenta, se copian los diferentes archivos de configuración desde el contenido del directorio /etc/skel (skeleton) hacia el directorio personal. Si desea modificar los entornos de manera global ANTES de crear los usuarios, puede colocar en **`/etc/skel`** todos los archivos que desea y modificarlos a su conveniencia. Así, por ejemplo, si desea que todo el mundo disponga de los mismos iconos por defecto en su escritorio y que tenga la misma configuración por defecto del escritorio, coloque en él los directorios **`Desktop`** y **`.kde`** de una cuenta modelo.

b. Scripts de configuración

El capítulo El shell y los comandos GNU le mostró los diferentes archivos cuyo contenido ejecuta el shell. Durante el proceso de inicio de sesión de un usuario se ejecutan los siguientes scripts, en este orden:

- **`/etc/profile`**: define las variables de entorno importantes como PATH, LOGNAME, USER, HOSTNAME, HISTSIZE, MAIL y INPUTRC.
- **`/etc/profile.d/*`**: /etc/profile invoca a todos los scripts presentes en ese directorio. Estos scripts pueden completar la configuración global añadiendo por ejemplo, la configuración de los parámetros de idioma, de los alias globales, etc.
- **`~/.bash_profile`**: si el shell es bash, se ejecutará el script siguiente. Está en el directorio de usuario y llama a otro script: **`~/.bashrc`**, que a su vez llama a**`/etc/bashrc`**. En .bash_profile puede definir variables adicionales. Por su parte, en**`~/.bashrc`** podrá determinar alias y funciones. No hay reglas estrictas.
- Se utiliza **`/etc/bashrc`** para definir las funciones y alias para todo el sistema y todos los usuarios en bash.

c. Grupos privados y setgid

En cuanto a la seguridad de los usuarios, la política de Red Hat y otras distribuciones asociadas como Fedora consiste en crear para ellos de manera sistemática un grupo privado y asignarles una máscara 002. Así, por defecto se protegen mejor los archivos, ya que ningún grupo puede acceder a los archivos de forma automática: los archivos tienen como grupo con privilegios uno que sólo incluye un miembro.

Dado que un usuario puede formar parte de varios grupos, en teoría puede extender el acceso a sus archivos y directorios con el comando **chgrp**, encargado de cambiar el grupo de un archivo. Del mismo modo, puede cambiar de manera temporal de grupo primario con el comando **newgrp**.

Sin embargo, el usuario suele preferir no "rayarse" y dar todos los permisos con un **chmod** (p. ej.: chmod 777), lo que constituye una brecha en la seguridad.

En este caso, la solución consiste en utilizar el bit **setgid** en un directorio.

- Se define un grupo común a todos los usuarios que deben poder acceder a un directorio y a sus archivos.
- Se crea un directorio cuyo grupo propietario es el grupo común.
- Se dan los derechos al directorio con el umask 002 (`rwxrwxr-x`) u otro, pero con todos los derechos sobre el grupo.
- Se añade al directorio el derecho setgid (s) sobre el grupo.

```
# mkdir dir
# chgrp grpcomun dir
# chmod 2770 dir
```

Cuando se ubica **setgid** en un directorio, todos los archivos creados en este directorio pertenecen al grupo del directorio, y no al grupo del usuario. Así, todos los miembros del grupo común podrán acceder a los datos. Sin embargo, el archivo sigue perteneciendo a su creador.

Asociando el Sticky Bit, se obtiene un buen nivel de protección y acceso a los datos:

- todos los miembros del grupo pueden crear en él archivos y directorios;
- todos los archivos pertenecen de manera automática al mismo grupo;
- sólo los propietarios de los archivos pueden suprimirlos.

```
# chmod 3770 dir
# ls -ld dir
drwxrws--T 2 seb grpcomun 1024 2008-04-24 11:39 dir/
$ ls -l test
-rw-r--r-- 1 seb grpcomun 0 2008-04-24 11:43 test
```

9. Resumen general de PAM

PAM (*Pluggable Authentication Modules*) es un conjunto de módulos y una librería que permiten instalar mecanismos de autenticación avanzados para todas las herramientas que necesitan una seguridad aumentada. La autenticación se basa en módulos. Cada módulo puede utilizar mecanismos diferentes para intentar autenticar a un usuario. Uno se basará en una autenticación Unix clásica; otro, en LDAP; un tercero, en Active Directory, y un último, en el reconocimiento de una firma digital. Este mecanismo permite también la comprobación mediante un dongle USB...

El principio es bastante sencillo; la configuración, más complicada. Un programa, que necesita confirmar una autenticación, llama a la librería **`libpam.so`**. Esta librería lee un archivo de configuración en el cual se puede requerir la invocación de varios módulos. Cada módulo al que se llama devuelve verdadero o falso. Según la configuración, verdadero puede ser un requisito para seguir con otro módulo, o ser suficiente para conectarse.

Para saber si un comando utiliza PAM, hay que mirar si está montado con la librería PAM:

```
$ ldd /usr/bin/passwd
        linux-vdso.so.1 =>  (0x00007fff4c14d000)
        libshadow.so.0 => /lib/libshadow.so.0 (0x00007f2e4072b000)
        libpam.so.0 => /lib/libpam.so.0 (0x00007f2e4051d000)
        libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x00007f2e40318000)
        libc.so.6 => /lib/libc.so.6 (0x00007f2e3ff91000)
        libcrypt.so.1 => /lib/libcrypt.so.1 (0x00007f2e3fd5a000)
        libdl.so.2 => /lib/libdl.so.2 (0x00007f2e3fb55000)
        /lib/ld-linux-x86-64.so.2 (0x00007f2e4096b000)
$ ldd /usr/bin/passwd|grep libpam >/dev/null && echo "Utiliza PAM"
Utiliza PAM
```

Se colocan los archivos de configuración en **`/etc/pam.d`**. Suele existir uno por aplicación que utiliza PAM y que lleva el mismo nombre. Si falta el archivo, se utiliza **`other`**. La sintaxis es la siguiente:

```
Type_module control módulo argumentos
```

Type_module

- **`auth`**: módulo de autenticación (por ejemplo, petición de login y contraseña).
- **`account`**: autorización, gestión de cuentas (verificación del usuario para el servicio dado.

¿Se autoriza?).

- **password**: comprobación y actualización de la información de seguridad (p. ej.: ¿sigue siendo válida la contraseña? Si no es así, petición de una nueva contraseña).
- **session**: modificación del entorno del usuario.

control

- **required**: éxito requerido. En caso de denegación, se llama a los módulos restantes a pesar de todo, pero, pase lo que pase, al final PAM devolverá una denegación.
- **requisite**: la denegación termina de manera inmediata la autenticación. El éxito le autoriza a proseguir.
- **sufficient**: un éxito evita los otros módulos. Dicho de otro modo, PAM devuelve ok pase lo que pase en caso de éxito.
- **optional**: se ignora el resultado.

módulo

- **pam_unix.so**: autenticación estándar mediante la función C getpw().
- **pam_env.so**: definición de las variables de entorno.
- **pam_securetty.so**: prohíbe una conexión de superusuario (root) desde un terminal sin suficiente seguridad. Se coloca la lista de los terminales autorizados en /etc/securetty.
- **pam_stack.so**: llama a otro servicio PAM para cargar módulos adicionales.
- **pam_nologin.so**: prohíbe la conexión de usuarios si el archivo **/etc/nologin** está presente. En ese caso, se visualiza su contenido.
- **pam_deny.so**: siempre devuelve una denegación.
- **pam_console.so**: da permisos adicionales a un usuario local.

Los parámetros dependen de cada módulo. Veamos un ejemplo en el caso clásico de un inicio de sesión por la consola. En este caso, el shell de conexión llama al comando **login**. ¡Ojo! Es sólo un ejemplo.

```
$ cat login
#%PAM-1.0
auth       requisite    /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_unix.so
auth       required     /lib/security/pam_deny
auth       required     /lib/security/pam_nologin.so
```

- La primera línea comprueba si root intenta conectarse desde un terminal sin seguridad (p. ej.: telnet, rlogin, rsh, etc.). Si es así, PAM devuelve directamente falso y la autenticación fracasa.
- La segunda línea carga el entorno del usuario. Un fracaso no impide aquí la ejecución de las líneas siguientes, pero al final PAM devolverá falso.
- La tercera línea intenta una autenticación mediante los mecanismos Unix clásicos (archivo de contraseñas, NIS, etc.). El éxito detiene el proceso en este punto: se autentica directamente al usuario. Si no, se pasa directamente a la línea siguiente.
- La cuarta línea siempre devuelve falso. Como los módulos de autenticación anteriores han fracasado, la conexión del usuario fracasa. A pesar de todo, se ejecuta la línea siguiente.
- Si el archivo **/etc/nologin** existe, se muestra.

Es posible prohibir el acceso a una lista de usuarios dados. Coloque los nombres de los usuarios prohibidos en **/etc/nologinusers** y añada la línea siguiente en el archivo de configuración PAM. En nuestro ejemplo, habría que añadirla entre la segunda y la tercera línea.

```
auth required /lib/security/pam_listfile.so onerr=suceed item=user
sense=deny file=/etc/nologinusers
```

La distribución Red Hat está configurada de tal manera que el archivo **/etc/security/system-auth** se invoca en casi todas las configuraciones PAM. Un ejemplo del archivo de configuración login es:

```
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
...
```

En este caso, usted hubiera puesto nuestra línea debajo de la de pam_nologin.so. Sin embargo, no debe olvidar que se utiliza el archivo de configuración **/etc/pam.d/login** únicamente para la autenticación desde el comando **login** y, por lo tanto, para una conexión desde un terminal (consola de texto). Desde una ventana de conexión gráfica (X-Window, x/k/gdm) no obtendrá el efecto deseado. En este caso, se tendría que haber hecho la modificación en **/etc/pam.d/system-auth**.

```
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        required      /lib/security/pam_deny.so
...
```

Tenga cuidado otra vez de colocar la línea en el sitio correcto. Después de este bloque, la línea no surtirá ningún efecto por culpa de la línea **sufficient** si el usuario tiene un login y una contraseña válidos. Por lo tanto, tendrá que colocar la línea al principio de las líneas **auth**.

La impresión

1. Principio

Existen tres estándares de impresión en Unix, uno en System V, otro en BSD y un último federador.

Sea cual sea el estándar, el principio de configuración es el mismo. A cada impresora detectada (en general en /etc/printcap) corresponde una cola de espera (**queue**). Un servicio independiente gestiona el conjunto de estas colas de espera. Este mecanismo permite la impresión multiusuario (los trabajos de impresión están en cola de espera, **job queues**) y en red (se puede usar el servicio desde otra máquina remota).

En general, todas las impresoras saber imprimir directamente texto llano ASCII en 80 columnas. Para imprimir documentos formateados o imágenes, se puede utilizar un driver. En realidad se habla de **filtro de impresión**. El filtro puede ser un script o un binario que recupera el flujo entrante (texto, imagen, documento, postscript...), lo identifica y, con la ayuda de tratamientos asociados, lo transforma en lenguaje comprensible por la impresora (PostScript, PCL, Canon, Epson, WPS...).

Si tiene la posibilidad y los medios, no dude en elegir una impresora compatible Postscript, que representa una garantía de compatibilidad perfecta. El sitio linuxprinting.org dispone de una base completa de compatibilidad de las impresoras en Linux.

Linux acepta los comandos procedentes de los Unix de tipo System V y BSD. Durante mucho tiempo, el subsistema de impresión se basaba en los servicios BSD y el demonio **lpd**. Desde hace unos años, todas las distribuciones se basan en CUPS, compatible (los comandos, al menos) con los antiguos sistemas de impresión.

2. System V

Los comandos de gestión de las colas de espera e impresiones con System V son los siguientes:

- `lp [-dImpresora] [-nCifra] fic1`: imprime el contenido del archivo fic1. La opción `-d` permite elegir la impresora; `-n`, el número de ejemplares.
- `lpstat [-d] [-s] [-t] [-p]`: información relativa a la impresión. La opción `-d`muestra el nombre de la impresora por defecto; `-s`, un estado resumido de las impresoras; `-t`, toda la información (estado, trabajos...); `-p [lista]`, únicamente la información sobre las impresoras incluidas en la lista.
- `cancel [ids] [printers] [-a] [-e]`: suprime las tareas de impresión ids de las impresoras printers. La opción `-a` suprime todos los trabajos del usuario; `-e`, todos los trabajos (únicamente para el administrador).
- Se pueden encontrar comandos **enable** y **disable** que emplean como parámetro el nombre de la cola de espera, lo que permite activar o desactivar su acceso.

El demonio suele llevar el nombre de **lpd** (*line printer daemon*) o **lpsched** (*line printer scheduler*).

- **lpadmin** permite administrar los servicios de impresión como las colas de espera asociadas a una impresora y la cola de espera por defecto. P. ej.: `lpadmin -p cola1 -v impresora-m modelo`.
- `lpadmin -x fila`: supresión de la cola de espera.
- `lpadmin -d fila`: definir la cola de espera por defecto.
- `lpadmin -p fila -u allow:lista`: autorización de imprimir para los usuarios especificados.

- `lpadmin -p fila -u deny:lista`: prohibición de imprimir para los usuarios especificados.
- `lpshut` detiene el servicio de impresión. Cuando se inicia de nuevo el demonio, se reanudan las impresiones en curso en el momento de la parada.
- `accept` y `reject` permiten validar una cola de espera para la impresión o cerrarla.
- `lpmove` permite transferir peticiones de impresión de una cola de espera hacia otra.

3. BSD

- `lpr [-dImpresora] [-#copias] fic1`: imprime el contenido del archivo fic1. La opción `-P` permite especificar la impresora; `-#`, el número de copias.
- `lpq [-Pimpresora]`: indica el estado y la lista de los trabajos para la impresora especificada si es preciso por la opción `-P`.
- `lprm [-Pimpresora] [-] [ids]`: permite suprimir un trabajo de la impresora especificada por la opción `-P`, la opción `-` suprime todos los trabajos del usuario, ids representa una lista de trabajos para suprimir.
- El comando **lpc** es una especie de pequeño shell que permite controlar las impresoras y los trabajos.

El servicio se suele llamar **lpd**.

4. CUPS

a. Presentación

CUPS (*Common Unix Printing System*) es un sistema de impresión de Unix orientado a red:

- Basado en el protocolo **IPP** (*Internet Printing Protocol*), que se basa a su vez en el protocolo**HTTP/1.1**.
- Fácil de utilizar, en particular gracias a una configuración y administración centralizada desde una interfaz HTTP, reglas de conversión basadas en los tipos MIME y archivos de descripción de impresora estándares (**PPD**, *PostScript Printer Description*).
- CUPS recoge los comandos System V y BSD ya tratados para más sencillez.
- Las trazas de las impresiones están disponibles en el formato **CLF** (*Common Log Format*) de servidores web y se pueden aprovechar con las mismas herramientas.
- CUPS es capaz de interactuar con los servidores de impresión LPD para guardar una compatibilidad ascendente.
- CUPS dispone de su propia API, que permite crear interfaces de usuario que pueden integrarse en entornos gráficos o interfaces de administración.
- Los pools de impresión permiten redireccionar automáticamente las tareas.
- La autenticación es posible por usuario, máquina o certificado numérico.

El servicio de impresión se llama **cupsd**.

```
$ ps -ef | grep cupsd
root      6924     1  0 11:20 ?        00:00:00 cupsd -C /etc/cups/cupsd.conf
```

No hacen falta herramientas gráficas para administrar un servidor CUPS. Sin embargo, para facilitar la administración de la impresión, la mayoría de las distribuciones cuentan con ellas. CUPS ofrece

una interfaz de administración WEB directamente accesible desde el puerto 631 del servidor. La interfaz funciona con cualquiera navegador HTTP.

```
http://localhost:631
```

El archivo de configuración es **/etc/cups/cupsd.conf**. El equivalente al archivo**/etc/printcap** se encuentra en **/etc/cups/printers.conf**.

```
<DefaultPrinter lj2100>
Info Laserjet 2100
DeviceURI socket://192.168.1.10:9100
State Idle
StateTime 1203806079
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
</Printer>
```

Si la página de inicio de CUPS no se muestra o el navegador le devuelve un mensaje de error, compruebe en el archivo **/etc/cups/cups.conf** el estado de las líneas siguientes que activan la interfaz web y los permisos de acceso:

```
# Only listen for connections from the local machine.
Listen localhost:631
Listen /var/run/cups/cups.sock

# Show shared printers on the local network.
Browsing On
BrowseOrder allow,deny
BrowseAllow all
BrowseLocalProtocols CUPS dnssd

# Default authentication type, when authentication is required...
DefaultAuthType Basic

# Web interface setting...
WebInterface Yes

# Restrict access to the server...
<Location />
  Order allow,deny
</Location>

# Restrict access to the admin pages...
<Location /admin>
  Order allow,deny
</Location>

# Restrict access to configuration files...
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
</Location>
```

b. Añadir una impresora

Tiene dos soluciones para añadir una impresora:

- Editar los archivos a mano.
- Pasar por la interfaz web o una herramienta de su distribución.

En el primer caso, debe modificar el archivo printers.conf para añadir una sección para su impresora y luego copiar en /etc/cups/ppd el archivo PPD correspondiente a su impresora. Después lo tiene que renombrar con el nombre de sección (p.ej.: lj2100.ppd) del archivo printers.conf. Finalmente, debe buscar la configuración de CUPS. En Red Hat u openSUSE, por ejemplo:

```
# service cups reload
```

Si la interfaz aparece en inglés, compruebe en el archivo de configuración que el parámetro DefaultLanguage alberga el valor es. También tiene que comprobar el lenguaje por defecto del navegador. Si a pesar de todo, el idioma de la interfaz sigue siendo el inglés, entonces compruebe si hay disponible un paquete de idiomas adicional para su distribución. En el peor de los casos, la interfaz sigue siendo comprensible (trabajos = jobs, impresoras = printers, etc.).

Como se suele activar la interfaz web por defecto, puede pasar por un navegador web. Es posible que durante el acceso a las páginas de administración se le requiera identificación. En principio, los de root son suficientes, pero puede crear cuentas (o añadir usuarios) encargados de la gestión de las impresiones con el comando **lppasswd**.

```
$ sudo lppasswd -a seb
Introduzca la contraseña:
Introduzca nuevamente la contraseña:
```

- En la página principal, haga clic en la pestaña **Administración**.

Interfaz de administración de CUPS

Es posible que CUPS detecte las impresoras locales o en red cuando éstas cuentan con servicios LPD (puerto 9100) o IPP, ya que el servicio está siempre a la escucha. En el caso de seleccionar la impresora entre las **Impresoras en red descubiertas** la tarea se facilita significativamente.

En este ejemplo, basado en añadir manualmente una impresora Samsung CLP 320, añadirá una impresora al hacer clic en **Añadir impresora**.

- La etapa siguiente consiste en elegir su tipo de conexión. La impresora está conectada en red. En este caso seleccione **AppSocket/HP JetDirect**. Haga clic en **Siguiente**.

Conexión de una impresora de red

Para una impresora en red, tiene varias opciones. La mayoría de las impresoras conectadas a una red de tipo Ethernet o WiFi ofrecen servicios de impresión LPD o Socket (impresión directa). En este último caso, es la propia impresora la que gestiona los trabajos de impresión. Para el resto (IPP, http, Samba, etc.), deberá consultar el manual de su impresora. Algunas impresoras profesionales ofrecen una interfaz de configuración web (como los routers que tenemos en casa) para configurar los puertos.

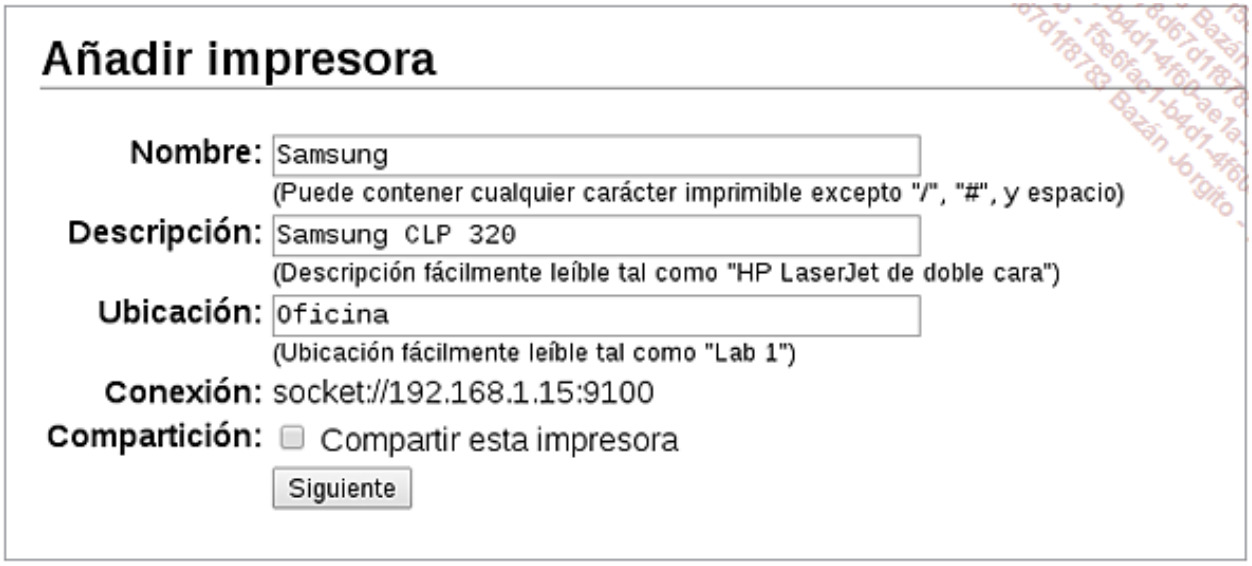
Si su impresora remota ya está configurada en otro servidor CUPS, puede pasar este servidor y las**URI** de tipo http, ipp o lpd para imprimir por ellos.

La impresora que usaremos en este ejemplo es la Laserjet 320 que dispone de una tarjeta de red y de un servidor de impresión jetDirect integrados. La URI introducida es: `socket://192.168.1.15:9100`

La IP es la de la impresora de red, el puerto 9100 es el puerto estándar en este caso.

A continuación haga clic en **Siguiente**.

- Inserte la información siguiente:
    - El **nombre** de la impresora es el nombre de la cola que será visible en las herramientas de impresión. No utilice espacios.
    - Se puede dejar la **ubicación** vacía, pero puede ser útil para localizar la impresora (en el caso de una impresora remota).
    - La **descripción** puede ser lo que usted desee, como por ejemplo el modelo completo de la impresora.



Nombre de la impresora

Haga clic en **Siguiente**.

Ahora elija un controlador que corresponda al modelo de su impresora. Observará que para varios modelos existen varios controladores. Se suele indicar el controlador aconsejado con**recommended**; tendrá que comprobar el controlador realmente recomendado para su impresora en el sitio siguiente:

http://www.linux-foundation.org/en/OpenPrinting

Que informa de que, para esta impresora:

For basic printing functionality use the PostScript PPD. For advanced functionality such as printer status and maintenance features, use the **HPLIP** driver (which includes **HPIJS**).

Por lo tanto, se debe elegir el driver **HP Laserjet 2100 hpijs**.

Linux es compatible con las impresoras (y los escáneres) de la marca HP gracias al proyecto libre del editor **HPLIP** (HP Linux Imaging and Printing), incluyendo las soluciones integradas (impresora, escáner, fax, fotocopiadora, impresión fotográfica, etc.). La lista del hardware compatible está disponible mediante http://hplip.sourceforge.net/. Por supuesto, otros fabricantes ofrecen impresoras y escáner perfectamente compatibles con Linux, como Samsung.

Es posible especificar otro archivo PPD. Varios fabricantes ofrecen este tipo de archivos para su impresora. Se ofrecen los archivos PPD por defecto en /usr/share/cups/model.

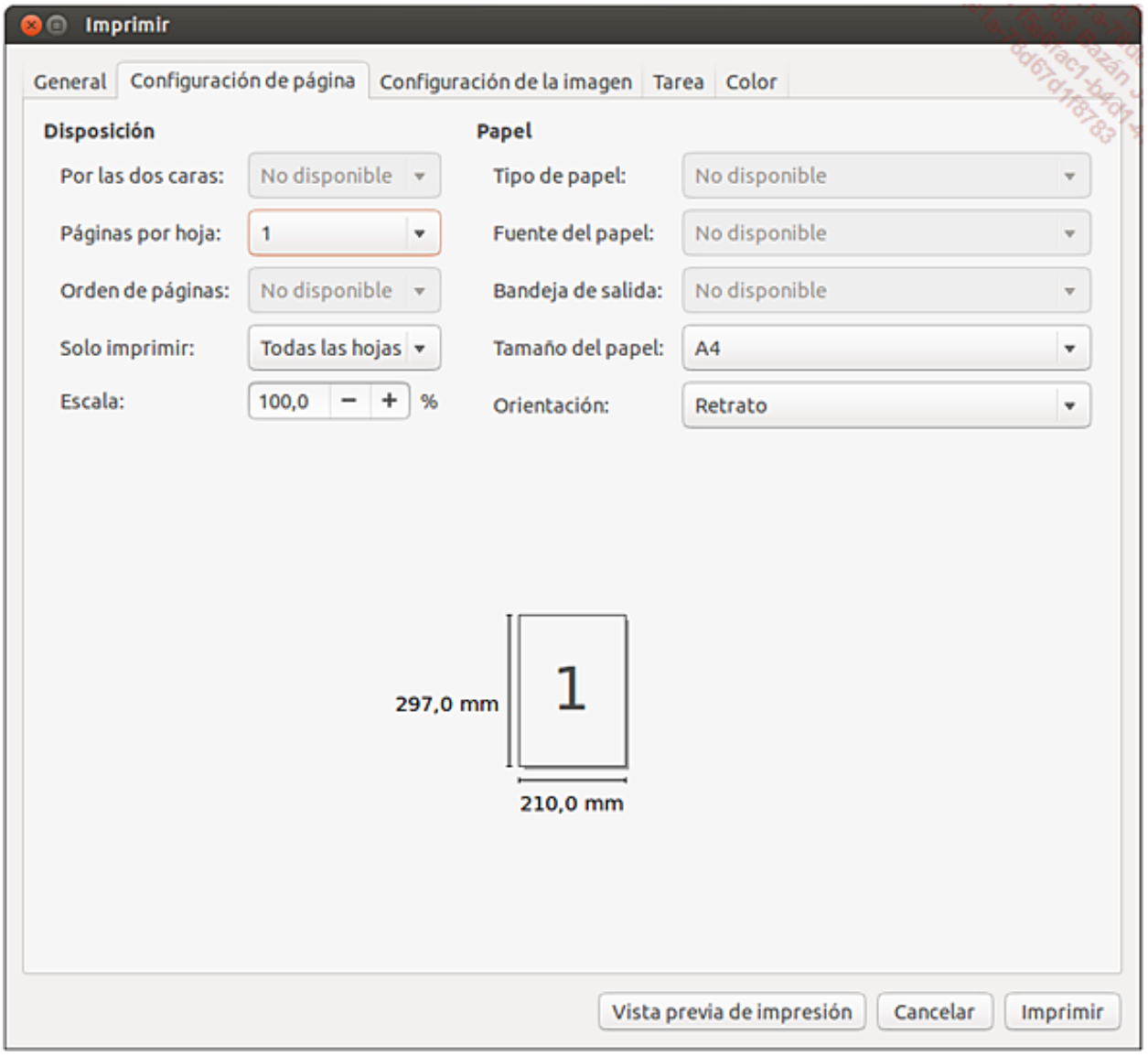
Selección del controlador correspondiente al modelo

- Haga clic en **Añadir impresora**.
- Llega a la pantalla que presenta la lista de las impresoras actualmente configuradas. Puede probar su impresora imprimiendo la página de prueba. También puede modificar las opciones por defecto de la impresora. Estas opciones son de varios tipos:
    - Opciones generales para indicar el tipo de papel por defecto (A4), la calidad de salida, la bandeja por defecto, la impresión a dos caras, el color, etc. Estas opciones pueden ser modificadas más tarde en el momento de la impresión empleando los cuadros de diálogo.
    - El modo de salida y las diferentes opciones, para ajustar por ejemplo la resolución y el tipo de papel por defecto, un filtro cualquiera, etc.
    - Las marcas de separación: para separar las hojas (en el caso de que se use papel continuo), puede ubicar la marca antes y después de la impresión. Se puede personalizar el archivo de marcas de impresión hasta cierto punto (vea el manual de CUPS para ello).
    - La facturación. Se puede configurar CUPS para gestionar la facturación por servicio/máquinas/usuarios, en el caso de que se utilicen algunos servicios avanzados.

Configuración de las opciones de la impresora

Tendría que dejar, además del formato de papel, las opciones por defecto. La mayoría de los entornos de escritorio y programas proponen modificar las opciones de impresión mediante CUPS, como en los sistemas de tipo Windows.

La captura siguiente fue efectuada en Gnome 3. En **Shotwell**, se llamó a la entrada **Imprimir** del menú **Archivo** durante la visualización de una imagen. Todas las opciones accesibles a través del botón **Propiedades** al lado del nombre de la impresora proceden de las posibilidades propuestas por CUPS y el driver de impresión PPD. Dos impresoras distintas, e incluso dos drivers distintos (si hay varios disponibles), no tendrán las mismas opciones.



La impresión en Gnome

Automatización

1. Con cron

a. Presentación

El servicio **cron** permite la programación de eventos repetidos. Funciona con ayuda de una tabla, llamada una **crontab**. Es un archivo de texto que se puede editar con un simple editor, como por ejemplo vi. Para modificar su crontab personal, utilice el comando **crontab** para editar la tabla con el parámetro `-e`.

Se guardan los archivos contrabs en `/var/spool/cron`.

El servicio **cron** debe estar ejecutándose para que las crontabs estén activas.

```
$ ps -ef|grep cron
root      3634     1  0 18:28 ?        00:00:00 /usr/sbin/cron
```

b. Formatos

El formato de un registro de crontab es el siguiente:

Minutos	Horas	Día del mes	Mes	Día semana	Comando
1	2	3	4	5	6

Utilice el formato siguiente para los valores periódicos:

- Un valor para indicar cuándo se debe ejecutar el comando. P. ej.: el valor 15 en el campo minuto significa el decimoquinto minuto.
- Una lista de valores separados por comas. P. ej.: 1,4,7,10 en el campo mes para enero, abril, julio, octubre.
- Un intervalo de valores. P. ej.: 1-5 en el campo día de la semana indica de lunes (1) a viernes (5). El 0 es el domingo y el 6, el sábado.
- El carácter * para todos los valores posibles. P. ej.: * en el campo día del mes indica todos los días del mes o de los meses.
- Una / indica un intervalo. Pe: */5 en el campo minutos indica cada 5 minutos.

c. Ejemplos

Ejecución de df todos los días, todo el año, cada cuarto de hora:

```
0,15,30,45 * * * * df > /tmp/libre
```

O

```
*/15 * * * * df > /tmp/libre
```

Arranque de un comando cada 5 minutos a partir de 2 (2, 7, 12, etc.) a las 18 horas los días 1 y 15 de cada mes:

```
2-57/5 18, 1, 15 * * comando
```

Ejecución de un comando todos los días laborables a las 17 horas:

```
0 17 * * 1-5 fin_trabajo.sh
```

Listar las crontabs activas:

```
$ crontab -l
```

Suprimir la crontab activa:

```
$ crontab -r
```

Editar la crontab de un usuario particular:

```
# crontab -u user
```

d. crontab sistema

La configuración crontab general para el sistema se encuentra en **/etc/crontab**. Su sintaxis difiere un poco. Hay un campo adicional. Permite especificar el usuario con el que se ejecutará el comando.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Aquí todos los días a las 4:02 h de la mañana se ejecuta **run-parts /etc/cron.daily**. El script **run-parts** acepta como parámetro un directorio y ejecuta todos los programas presentes en éste.

```
$ ls cron.daily/
00-logwatch  0anacron   makewhatis.cron  slocate.cron
00webalizer  logrotate  rpm              tmpwatch
```

Entre los programas ejecutados, fíjese en **logrotate**, que permite efectuar copias de seguridad y renombrar archivos logs y archivos de diario del sistema para que éstos no se vuelvan inabarcables a consecuencia de su tamaño. El programa **tmpwatch** se encarga de limpiar el sistema de los archivos ociosos (en /tmp por ejemplo).

Para terminar, el directorio **/etc/cron.d** contiene crontabs adicionales siempre en formato crontab sistema. Por lo tanto, una aplicación que debe ejecutar tareas recurrentes deberá crearse su propio archivo en vez de modificar una tabla existente.

e. Control de acceso

Se puede controlar el acceso con el comando **crontab** por usuario con los archivos **/etc/cron.allow** y **/etc/cron.deny**.

- Si cron.allow está creado, sólo los usuarios explícitamente indicados pueden utilizar **cron**(ver Automatización - Con at, en este capítulo).
- Si cron.allow no está creado, cron comprueba la presencia de un archivo cron.deny. Todos los usuarios que no estén en él están autorizados a utilizar cron. Si está vacío, todo el mundo está autorizado a utilizar **cron**.

- Si no existe ninguno de los dos archivos, sólo root puede utilizar cron.

2. Con at

a. Presentación

El comando **at** y los comandos asociados permiten la gestión de los batchs. A diferencia de la contrab, las modificaciones son volátiles: se pierden cuando se termina la sesión. Le corresponde a usted colocar la lista de los comandos en un posible archivo y cargarlo si es necesario mediante los scripts de su perfil.

Para que funcione at, el servicio **atd** (*at daemon*) debe estar en marcha.

```
$ ps -ef | grep atd
at        7988     1  0 21:05 ?        00:00:00 /usr/sbin/atd
```

b. Formatos

Para simplificar, hay dos maneras de utilizar at:

- pasándole una línea de comandos de manera interactiva,
- pasándole un archivo ejecutable que contiene los comandos que se deben ejecutar.

En los dos casos, debe facilitar a at una hora de ejecución. El formato de esta hora es bastante flexible. Para programar la ejecución de una línea de comandos a las 21:20 h de manera interactiva:

```
$ at 21:20
warning: commands will be executed using /bin/sh
at> echo hola
at> <EOT>
job 4 at 2008-05-08 21:20
```

Después de haber introducido el comando o los comandos que se deben ejecutar a las 21:20 h, pulse [Entrar], y en una línea vacía pulse [Ctrl] **D** (final de entrada). El comando **at** confirma la programación del comando.

Para programar la ejecución de un comando (script o binario) a las 21:25 h:

```
$ at -f /home/seb/test.sh 21:25
warning: commands will be executed using /bin/sh
job 6 at 2008-05-08 21:25
```

Hora

Se puede formatear la hora de la manera siguiente:

- HHMM o HH:MM.
- La hora puede tener el formato de 12 o 24 h. Con el formato de 12 horas, puede especificar AM (mañana) o PM (tarde).
- Midnight (medianoche), noon (mediodía), teatime (16:00 h, típicamente inglés).
- MMJJAA, MM/JJ/AA o JJ.MM.AA para una fecha particular.
- Now: ahora.
- + n minutes/hours/days/weeks: la hora actual a la que se añaden n minutos/horas/días/semanas.

Si la hora especificada es inferior a la hora actual, se ejecuta el comando al día siguiente.

```
$ at 21:30 09.05.2008
warning: commands will be executed using /bin/sh
at> echo ¡hola!
at> <EOT>
job 9 at 2008-05-09 21:30
$ at now + 2 days
warning: commands will be executed using /bin/sh
at> echo en dos días
at> <EOT>
job 10 at 2008-05-10 21:29
```

También existe el comando **batch**, que no tiene en cuenta la hora. Ejecuta el comando en cuanto el nivel de trabajo de la máquina lo permite. Se puede especificar la hora, pero en este caso se considera como «a partir de esta hora en cuanto sea posible».

c. Control de las tareas

El comando **atq** (*at queue*) permite listar las tareas programadas:

```
$ atq
10      2008-05-10 21:29 a seb
9       2008-05-09 21:30 a seb
```

Se colocan los jobs (tareas) en el directorio `/var/spool/atjobs`, a razón de un ejecutable por tarea.

```
# ls -l /var/spool/atjobs/
-rwx------ 1 seb users 5620 may  8 21:29 a000090133cf92
-rwx------ 1 seb users 5628 may  8 21:30 a0000a0133d531
```

Si mira el contenido del ejecutable, verá que su comando no está solo. Está situado al final, pero el script ubica todo el entorno durante la creación de la entrada at.

```
#cat a0000a0133d531
#!/bin/sh
# atrun uid=1000 gid=100
# mail      seb 0
umask 22
LESSKEY=/etc/lesskey.bin; export LESSKEY
NNTPSERVER=news; export NNTPSERVER
INFODIR=/usr/local/info:/usr/share/info:/usr/info; export INFODIR
MANPATH=/usr/local/man:/usr/share/man:/opt/gnome/share/man; export
MANPATH
KDE_MULTIHEAD=false; export KDE_MULTIHEAD
... (unas 80 líneas) ...
cd /home/seb || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
echo ¡hola!
```

El comando **atrm** permite suprimir una tarea:

```
$ atrm 10
$ atrm 9
$ atq
```

d. Control de acceso

Es posible controlar el acceso al comando **at** por usuario con los archivos **/etc/at.allow** y **/etc/at.deny**.

- Si at.allow está creado, sólo los usuarios explícitamente indicados pueden utilizar **at**.
- Si at.allow no está creado, at comprueba la presencia de un archivo at.deny. Todos los usuarios que no están en él están autorizados a utilizar at. Si está vacío, todo el mundo está autorizado a utilizar el comando at.
- Si no existe ninguno de los dos archivos, sólo root puede utilizar **at**.

3. Con anacron

El comando **anacron** ejecuta periódicamente los comandos con una frecuencia especificada en días. A diferencia de cron, el equipo no debe funcionar las 24h. Anacron puede remplazar a cron en equipos que se apagan con frecuencia (estaciones de trabajo, equipo casero, etc.): los comandos se ejecutan cuando el tiempo es alcanzado o sobrepasado.

La configuración de anacron está ubicada en **/etc/anacrontab** y tiene el formato siguiente:

```
Periodo     espera     identificador     comando
```

Por ejemplo:

```
1 5 cron.daily nice run-parts --report /etc/cron.daily
7 10 cron.weekly nice run-parts --report /etc/cron.weekly
@monthly 15 cron.monthly nice run-parts --report /etc/cron.monthly
```

La segunda línea implica que cada 7 días **cron.weekly** debe ser ejecutado. Al arrancar, anacron verifica la fecha de la última ejecución. Si no han transcurrido más de siete días, **cron.weekly** se arranca 10 minutos más tarde (espera en minutos para evitar cualquier arranque simultáneo).

Los registros (logs) del sistema

1. Fundamentos

Cuando el sistema se inicia, se pone en marcha y efectúa cualquier tipo de acción, se registran sus acciones y las de la mayoría de sus servicios en diferentes archivos. Dos servicios están especializados en la recepción de los mensajes que tienen como destino estos archivos:

- **klogd**: *kernel log daemon*, encargado de la gestión de la información emitida por el núcleo.
- **syslogd**: *system log daemon*, encargado de la gestión de la información emitida por cualquier tipo de servicio y, si procede, del núcleo.

Ciertas distribuciones utilizan ahora **syslog-ng** o **rsyslog**, cuyas reglas de tratamiento de los mensajes, basadas en expresiones regulares, han evolucionado considerablemente. Sin embargo, el principio sigue siendo exactamente el mismo.

Desde un punto de vista histórico, el servicio syslogd gestionaba también los mensajes emitidos por el núcleo. Todavía es capaz de hacerlo, pero la cantidad de mensajes emitidos, los diferentes niveles de gravedad y los nuevos métodos de acceso a los mensajes del núcleo hacen que parezca importante y pertinente separar la gestión de los mensajes del núcleo y la de los emitidos por los servicios.

Los mensajes importantes emitidos por un componente del sistema deberían pasar por el servicio syslogd. Esto no impide, al contrario, que un servicio pueda gestionar sus propias entradas en sus propios archivos. No se deberían colocar entradas de aplicaciones en el registro del sistema. Las trazas de acceso a las páginas web de un servidor Apache no tienen nada que hacer ahí. En cambio, las de conexión al sistema (mediante la consola, ssh, telnet, etc.) tienen mucho interés y deben estar presentes en los archivos logs del sistema.

A partir de este punto y en el resto del libro, las entradas en el registro se llamarán por su nombre de uso corriente: los **logs**.

2. Los mensajes

El servicio **klogd** gestiona los mensajes emitidos por el núcleo. Dispone de dos fuentes de acceso a los mensajes:

- el sistema de archivos virtual **/proc**, utilizado por defecto si está presente, y en particular `/proc/kmsg`;
- las llamadas al sistema mediante la API del núcleo, en particular `sys_syslog`, si **/proc** no está presente o si se ha pasado el parámetro `-s` a klogd.

Los mensajes del núcleo tienen niveles de prioridad diferentes, escalados de 0 (alta prioridad) a 7 (mensaje de depuración):

Nivel	Alias sistema	Significado
0	**EMERG**	No se puede utilizar el sistema.
1	**ALERT**	Se debe efectuar una acción inmediatamente.
2	**CRIT**	Problema crítico.
3	**ERR**	Error.
4	**WARNING**	Aviso.
5	**NOTICE**	Normal, pero necesita una atención particular.

6	**INFO**	Información estándar.
7	**DEBUG**	Traza de depuración del núcleo.

El servicio **klogd** devuelve los mensajes de nivel 0 a 6 a **syslogd**, que los redirigirá a los archivos de logs y si es preciso a las consolas correspondientes. Por defecto, no se traza la información de depuración de nivel 7.

El servicio **syslogd** (o syslog-ng) recibe los mensajes procedentes de los servicios y también de klogd. Luego los reparte según el transmisor, la gravedad, en archivos, en consolas, en forma de mails a los usuarios del sistema (root, por ejemplo), etc.

Las acciones más corrientes son la escritura de los logs en archivos, la redirección de mensajes hacia una consola (muchas veces se trata de la 10 o 12) o el envío de mensajes a root.

3. Configuración de syslog

El archivo de configuración **`/etc/syslog.conf`** permite definir el origen, la importancia y el destino de cada mensaje, en forma de dos campos.

- El origen define de hecho un conjunto de **sistemas** y de **subsistemas** (núcleo, servicios). La lista, extensible, se compone en el origen de los elementos siguientes. El asterisco define el conjunto de los subsistemas.

Subsistemas	Significado
auth/authpriv	Servicio de seguridad y autenticación.
cron	Servicio cron.
daemon	Los demonios del sistema.
kern	El núcleo.
lpr	El servicio de impresión.
mail	La mensajería.
news	La red.
syslog	El propio Syslog.
user	Mensajes de los procesos de los usuarios.
uucp	Unix to Unix CoPy.
local0->7	Mensajes procedentes de klogd, la cifra representa el nivel.

- La importancia o **nivel** define el nivel de gravedad del mensaje. El asterisco define el conjunto de todos los niveles. Los niveles emitidos por **klogd** y **syslogd** son equivalentes.

Nivel	Significado
emerg	No se puede utilizar el sistema.
alert	Una intervención inmediata es indispensable.
crit	Error crítico para el subsistema.
err	Error de funcionamiento.
warning	Aviso.
notice	Evento normal que merece ser señalado.
info	Sólo como información.
debug	Mensaje mandado para el ajuste.

none	Ignorar los mensajes.

El destino o la **acción** puede ser un archivo, un mensaje a un usuario, la consola, una lista de usuarios... El asterisco indica todo el mundo.

Se inscriben los mensajes syslog en los archivos **/var/log/messages** y **/var/log/syslog** o en cualquier otro archivo configurado en **/etc/syslog.conf**.

El ejemplo siguiente procede de una instalación Red Hat AS 4u6.

```
# Se coloca todo (salvo mail.*) en /var/log/messages
*.info;mail.none;authpriv.none;cron.none     /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                   /var/log/secure

# Mails
mail.*                                       -/var/log/maillog

# Crontab
cron.*                                       /var/log/cron

# Mensajes de alerta
*.emerg                                      *

# errores uucp y news
uucp,news.crit                               /var/log/spooler

# Mensajes de boot
local7.*                                     /var/log/boot.log
```

Usted mismo puede mandar, directamente o en sus scripts, mensajes a **syslogd** mediante el comando **logger**.

4. El caso de rsyslog

El gestor de registros **rsyslog** se está usando cada vez más como sustituto de syslog y syslog-ng. Su configuración principal está en **/etc/rsyslog.conf**. Las configuraciones adicionales están en**/etc/rsyslog.early.conf**, utilizado antes de la activación de las capas de red, y el contenido de **/etc/rsyslog.d/*** se carga con el archivo de configuración principal.

Además de la sintaxis clásica de syslog, rsyslog dispone de módulos y de una sintaxis más completa. Esta sintaxis permite especialmente redirigir los mensajes en función de reglas más complejas que pueden describirse con la ayuda de comandos de programación. La redirección de los mensajes no se simplifica sólo en función de los niveles y subsistemas, sino que también analiza el contenido de los mensajes. La siguiente configuración tiene como objetivo redirigir los mensajes provenientes de la aplicación Network Manager a su propio archivo de registro. Lo hará a partir del nombre o de la porción de nombre del programa emisor.

```
if     ($programname == 'NetworkManager') or \
       ($programname startsWith 'nm-') \
then   -/var/log/NetworkManager
&      ~
```

Los archivos de syslog son compatibles con rsyslog, pero no hay reciprocidad: syslog no puede leer la sintaxis extendida de rsyslog. Se puede iniciar rsyslog en modo compatibilidad con el parámetro **-c0**.

5. Los registros

Por convención se ubican los logs del sistema en **/var/log**. No todos los logs de este directorio provienen de **syslogd**. Es el caso, por ejemplo, de la información de conexión. Veamos un ejemplo del contenido de este directorio. Contiene varios archivos de texto y directorios. Unos servicios pueden decidir, sin pasar por **syslogd**, concentrar y escribir sus mensajes en esta estructura.

```
# cd /var/log ; ls -l
-rw-r-----  1 root  root      2460 feb  7 05:34 acpid
drwxr-x---  2 root  root      4096 mar  5  2007 audit
-rw-------  1 root  root       116 mar 27 04:02 boot.log
-rw-------  1 root  root     75487 mar 28 11:10 cron
drwxr-xr-x  2 lp    sys       4096 mar 27 04:02 cups
-rw-r--r--  1 root  root     28359 feb  7 05:34 dmesg
drwx------  2 root  root      4096 ago  7  2007 httpd
-r--------  1 root  root  18747276 mar 28 11:08 lastlog
drwxr-xr-x  2 root  root      4096 jun  1  2007 mail
-rw-------  1 root  root      4537 mar 28 04:02 maillog
-rw-------  1 root  root    178348 mar 28 11:10 messages
drwx------  2 root  root      4096 oct 16 23:21 samba
-rw-------  1 root  root    214999 mar 28 11:08 secure
-rw-r--r--  1 root  root      2734 mar 28 11:01 snmpd.log
-rw-------  1 root  root         0 mar 23 04:02 spooler
drwxr-x---  2 squid squid     4096 ene 22  2007 squid
-rw-------  1 root  root     62165 mar 28 09:13 sudo.log
drwxr-xr-x  2 root  root      4096 oct  5  2004 vbox
-rw-rw-r--  1 root  utmp    127872 mar 28 11:10 wtmp
-rw-------  1 root  root     40557 mar 28 11:03 xferlog
```

6. Journald

journald es un servicio de systemd que permite recopilar y almacenar trazas. Las mantiene estructuradas e indexadas. El núcleo envía las trazas, por un servicio syslog, por los canales estándar o empleando una API. El comando **journactl** permite acceder a las trazas almacenadas por journald.

Cuando se emplea journald, las trazas y eventos son redirigidos a este. Para que rsyslog pueda leerlas y funcionar como habitualmente, se debe conectar a un socket ofrecido por journald.

Por ejemplo en Fedora:

```
# cat /etc/rsyslog.d/listen.conf
$SystemLogSocketName /run/systemd/journal/syslog
```

A continuación se presentan varios ejemplos de uso del comando **journactl**.

Ver todos los registros:

```
# journalctl
```

Esperar a nuevos registros con un equivalente a tail `-f`:

```
# journalctl -f
```

Mostrar las trazas según el comando que las ha enviado. Si toca la tecla [Tab] después de introducir _COMM, el completado automático le ofrecerá una lista de los comandos correspondientes, o completa lo que haya empezado a introducir.

```
# journalctl _COMM=sshd
-- Logs begin at mar. 2014-07-08 22:40:06 CEST, end at mer. 2014-07-09 22:22:18 CEST.-
juil. 08 22:40:10 localhost.localdomain sshd[523]: Could not load host key:
/etc/ssh/ssh_host_dsa_key
jul. 08 22:40:10 localhost.localdomain sshd[523]: Could not load host key:
/etc/ssh/ssh_host_ecdsa_key
jul. 08 22:40:10 localhost.localdomain sshd[523]: Server listening on 0.0.0.0 port 22.
```

```
jul. 08 22:40:10 localhost.localdomain sshd[523]: Server listening on :: port 22.
```

Si pulsa la tecla [Tab] después de introducir el caracter ” _ ”, el auto completado se activa y ofrece todas las posibilidades. Esto es válido para cada etapa de la introducción de parámetros en journalctl.

```
# journalctl _SYSTEMD_UNIT=crond.service
-- Logs begin at mar. 2014-07-08 22:40:06 CEST, end at mer. 2014-07-09 22:22:18 CEST. -
jul. 08 22:40:09 localhost.localdomain crond[354]: (CRON) INFO (Syslog will be used
instead of sendmail.)
jul. 08 22:40:09 localhost.localdomain crond[354]: (CRON) INFO (RANDOM_DELAY will be
scaled with factor 25% if used.)
jul. 08 22:40:09 localhost.localdomain crond[354]: (CRON) INFO (running with inotify
support)
```

Para terminar, puede modificar el format de salida con -o:

```
# journalctl _SYSTEMD_UNIT=crond.service -o json
```

Archivado y backup

1. Las herramientas de copia de seguridad

La copia de seguridad constituye un trabajo importante del administrador, ya que, en caso de problema grave, se suele restaurar el sistema desde una copia de seguridad, o una imagen del sistema cuando éste estaba todavía bien (funcionamiento correcto, sin corrupción). Se entrega cada Unix con sus propios comandos y procedimientos de copia de seguridad; sin embargo, existen algunas herramientas comunes.

a. Comandos, planes, scripts

- Para la copia de seguridad de los archivos y estructuras, utilice los comandos **tar** y **cpio**. Estos comandos guardan una estructura, y no un sistema de archivos. Se puede hacer coincidir los dos.
- Para la copia de seguridad física de discos y sistemas de archivos (los dumps), utilice el comando **dd**.

Una copia de seguridad incremental consiste en salvaguardar una primera vez la totalidad de los datos y luego únicamente los archivos modificados. También se pueden encontrar en forma de programas libres o en el mercado soluciones más avanzadas de copia de seguridad (Networker, por ejemplo).

A veces el administrador tendrá que definir scripts de copia de seguridad y restauración adaptados a cada caso concreto (partición del sistema, datos aplicativos...) y automatizar cuando sea posible la ejecución de estos scripts en función de la fecha, hora o carga de la máquina.

También será importante definir un plan de copia de seguridad y, para ello, hay que hacerse las preguntas correctas:

- ¿Qué se debe guardar?
- ¿Con qué frecuencia?
- ¿Cuánto tiempo se conservarán las copias de seguridad, dónde, cuántos ejemplares?
- ¿Dónde se almacenará el historial de las copias de seguridad?
- ¿Cuál es el soporte más apropiado?
- ¿Cuánta capacidad necesita el soporte de copia de seguridad?
- ¿Cuánto tiempo se prevé para guardar un archivo, un sistema de archivos?, y ¿es razonable?
- ¿La copia de seguridad debe ser automática o manual?
- ¿Cuál es el método más apropiado para la copia de seguridad?

Como cada caso es único, este libro no puede contestar a todas estas preguntas. Las respuestas dependen del entorno de destino (producción, integración, tests, etc.). Sin embargo, no se olvide de hacer una copia de seguridad del sistema (raíz, /opt, /usr, /var, /boot, etc.) después de una instalación y antes de una modificación importante, por si acaso fuera necesario volver atrás.

b. Otros comandos

- **mt**: control de una cinta magnética.
- **touch**: pone la fecha de última modificación a la hora actual, para forzar una copia de seguridad incremental.
- **find**: selecciona los archivos que hay que salvaguardar.

- **compress** y **uncompress**: compresión y descompresión de los archivos.
- **gzip**, **gunzip**, **zcat**, compresión y descompresión en formato GnuZip.
- **xz**, **unxz xzcat**: compresión y descompresión del formato .xz (derivado de lzma).

2. tar

El comando **tar** es simple y eficaz. Crea archivos de los archivos, incluida la estructura de archivos, en cualquier tipo de soporte, incluido en otro archivo (archivo con la extensión .tar). El archivo así creado puede extenderse a varios volúmenes: cuando la cinta o el disquete está lleno, le corresponde al usuario insertar uno nuevo. El proceso de copia/restauración continúa.

a. Archivadores

La sintaxis es la siguiente:

```
tar cvf nombre_archivo archivo(s)
```

Por ejemplo, para colocar en un archivo **tar** el directorio Desktop:

```
$ tar cvf desktop.tar Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

Los parámetros son los siguientes:

- `c`: creación de archivo;
- `V`: modo verboso, **tar** muestra lo que hace;
- `f`: el parámetro siguiente es el nombre del archivo.

b. Listar

La sintaxis es:

```
tar tvf nombre_archivo
```

Para listar el contenido del archivo anterior:

```
$ tar tvf desktop.tar
drwx------ seb/users       0 2008-04-17 09:44 Desktop/
-rw-r--r-- seb/users     191 2007-10-20 20:10 Desktop/fusion-icon.desktop
-rw-r--r-- seb/users    4786 2007-09-26 00:43 Desktop/konsole.desktop
-rw-r--r-- seb/users     665 2008-04-08 15:14 Desktop/Support.desktop
-rw-r--r-- seb/users    1051 2007-10-05 10:16 Desktop/Office.desktop
-rw-r--r-- seb/users    4586 2007-12-05 11:37 Desktop/Terminal.desktop
-rw-r--r-- seb/users     829 2007-10-17 12:12 Desktop/MozillaFirefox.desktop
```

```
-rw-r--r-- seb/users    3952 2007-10-05 10:16 Desktop/Printer.desktop
-rw-r--r-- seb/users    2053 2007-10-05 10:16 Desktop/.directory
-rw-r--r-- seb/users     450 2007-10-23 11:58 Desktop/myComputer.desktop
-rw-r--r-- seb/users     218 2008-02-22 08:43 Desktop/trash.desktop
-rw-r--r-- seb/users     328 2008-04-08 15:14 Desktop/SuSE.desktop
-rw-r--r-- seb/users     472 2008-04-17 09:44 Desktop/Windows.desktop
```

El parámetro `t` lista el contenido del archivo.

c. Restauración

Para restaurar el contenido de un archivo, la sintaxis es:

```
tar xvf nombre_archivo archivos
```

Para restaurar el archivo anterior:

```
tar xvf desktop.tar
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

El parámetro `x` permite la extracción del conjunto de los ficheros del archivo, o de los ficheros especificados después del nombre del archivo.

d. Otros parámetros

El comando **tar** de gnu permite gestionar los formatos de compresión directamente:

- z: se comprime el archivo al formato gzip.
- Z: se comprime el archivo al formato compress.
- j: se comprime el archivo al formato bzip2.
- J: se comprime el archivo en el formato xz.

Así, los comandos anteriores para el formato de compresión **gzip** se convierten en:

```
$ tar cvzf desktop.tar.gz Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
```

```
Desktop/Windows.desktop
$ ls -l desktop.tar*
-rw-r--r-- 1 seb users 30720 may  9 11:16 desktop.tar
-rw-r--r-- 1 seb users  7556 may  9 11:22 desktop.tar.gz
```

Observe la diferencia de tamaño. Se pueden utilizar las opciones de compresión con `c`, `t` y `x`. Observe que se comprime el archivo final, y no los archivos de manera individual. Puede ser preferible no especificar una opción de compresión si está haciendo una copia en un soporte que gestiona por sí mismo la compresión.

Si comprime su archivo para destinarlo a otros sistemas, o desea guardar una compatibilidad con los parámetros por defecto de tar, puede proceder así:

```
$ gzip -cd desktop.tar.gz | tar xvf -
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
```

El parámetro `-d` precisa a **gzip** que debe descomprimir el archivo, mientras que `-c` pasa el resultado por la salida estándar. El - final indica a tar que debe recuperar el flujo por la entrada estándar.

Observe que las versiones recientes de tar reconocen directamente el formato de compresión, si se emplea sin pasar las opciones específicas de estos formatos.

```
# file test.tar.xz
test.tar.xz: XZ compressed data
# tar xf test.tar.xz
# echo $?
0
```

3. cpio

El comando **cpio** hace una copia de seguridad sobre la salida estándar de los archivos cuyos nombres están insertados en la entrada estándar, por defecto la pantalla y el teclado. Por lo tanto, es posible utilizar las redirecciones. **cpio** no comprime los archivos. Le corresponde a usted hacerlo.

a. Archivar

La sintaxis general es:

```
cpio -oL
```

Los parámetros más utilizados son:

- `-o`: output, creación de la copia de seguridad en salida.
- `-L`: copia los archivos relacionados, y no los vínculos simbólicos.
- `-v`: modo verboso, información detallada.
- `-c`: copia atributos de los archivos en forma ASCII (para el intercambio entre varios SO).

Veamos cómo archivar y comprimir el directorio Desktop:

```
find Desktop -print| cpio -ocv | gzip > archive.cpio.gz
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
> ls -l archive.cpio.gz
-rw-r--r-- 1 seb users 7377 may  9 11:33 archive.cpio.gz
```

b. Listar

La sintaxis general es:

```
cpio -it archivo
```

Los parámetros son:

- `-i`: lectura del archivo en entrada.
- `-t`: como para tar, lista el contenido del archivo.

```
$ cat archive.cpio.gz | gzip -cd | cpio -it
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
```

c. Restaurar

La sintaxis general es:

```
cpio -i[umd]
```

- `-u`: restauración incondicional, con borrado de los archivos que ya existen. Por defecto no se restauran los archivos si los que están presentes en el disco tienen como fecha de última modificación la misma o posterior.
- `-m`: los archivos restaurados conservan su última fecha de modificación.
- `-d`: cpio reconstruye el árbol de directorios y subdirectorios que faltan.

Para restaurar el archivo anterior:

```
$ cat archive.cpio.gz | gzip -cd | cpio -iuvd
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
```

4. dd

El comando **dd** (*device to device*) tiene como misión la copia física, bloque por bloque, de un archivo periférico hacia un archivo periférico. Al principio se utilizaba para la lectura y escritura en cinta magnética, pero se puede utilizar también con cualquier archivo. El comando **dd** permite realizar copias físicas de discos y sistemas de archivos.

Argumento	Función
if=archivo	Nombre del archivo a la entrada (el que se debe copiar)
of=archivo	Nombre del archivo a la salida.
bs=n	Tamaño del bloque en bytes.
count=n	Número de bloques para copiar.
skip=n	Número de bloque que hay que saltar al principio del archivo de entrada.
conv=	Conversión de la entrada.
seek=	Número de bloques que hay que saltar al principio del archivo de salida.
-s	Shell (comando de conexión) por defecto del usuario (variable SHELL). El usuario puede cambiarlo mediante el comando **chsh**.
-p	La contraseña del usuario. ¡Cuidado! La contraseña debe estar ya cifrada. Así, a menos que se recopie la contraseña de una cuenta genérica, usted preferirá utilizar luego el comando **passwd**.

La opción **conv** admite los parámetros siguientes:

- `ascii`: convertir el EBCDIC en ASCII.
- `ebcdic`: convertir el ASCII en EBCDIC.
- `block`: completar los bloques que terminan por un salto de línea con espacios, hasta alcanzar el tamaño mencionado por bs.
- `unblock`: sustituir los espacios al final de bloques (de tamaño cbs) por un salto de línea.
- `lcase`: transformar las mayúsculas en minúsculas.
- `ucase`: transformar las minúsculas en mayúsculas.
- `noerror`: seguir incluso después de errores de lectura.
- `notrunc`: no limitar el tamaño del archivo de salida.
- `sync`: completar cada bloque leído con ceros para alcanzar el tamaño ibs.

Aquí colocará el sector de boot MBR de la partición (donde está lilo o grub) en un archivo. El archivo así creado se podrá utilizar con el cargador de NT/2000/XP para un arranque en Linux.

```
# dd if=/dev/sda1 of=boot.lnx bs=446 count=1
```

Para crear un archivo vacío con un tamaño de 1 MB:

```
$ dd if=/dev/zero of=vacio bs=1024
count=1024
1024+0 registros leídos
1024+0 registros escritos
1048576 bytes (1,0 MB) copied, 0,0199192 s, 52,6 MB/s
```

El reloj

1. Conocer la hora

a. fecha

Para conocer la hora, utilice el comando **date**. Da la fecha actual, pero también calcula otras fechas en función o de la fecha actual o de cualquier fecha. **date** también modifica la fecha y hora del sistema.

```
$ date
sab may 10 13:58:38 CEST 2008
```

Por defecto, la fecha visualizada es la fecha (y hora) local, configurada en función del huso horario. Para visualizar la hora UTC:

```
$ date --utc
sab may 10 11:01:10 UTC 2008
```

Se puede modificar el formato de la fecha a voluntad, tal y como se puede hacer con la función C strftime. En este caso, la sintaxis es:

```
fecha +"formato"
```

Veamos algunos ejemplos de formatos posibles:

Formato	Resultado
%H	La hora tiene el formato 00..23.
%M	Minutos 00..59.
%S	Segundos 00..60.
%T	Hora actual sobre 24 horas.
%r	Hora actual sobre 12 horas.
%Z	Huso horario.
%a	Día abreviado (lun, mar, etc.).
%A	Día completo.
%b	Mes abreviado.
%B	Mes completo.
%d	Día del mes.
%j	Día del año.
%m	Número del mes.
%U	Número de la semana 00..53.
%y	Dos últimas cifras del año.
%Y	Año completo.

Para visualizar una fecha completa:

```
$ date +"Hoy es %A, %d %B %Y, son las %H horas, %M minutos y %S segundos"
Hoy es jueves, 10 de julio del 2014, son las 14 horas, 10 minutos y 20 segundos
```

Puede modificar la base de cálculo con el parámetro `--date` seguido de una fecha o de un cálculo. Se aceptan las palabras clave today, yesterday, tomorrow, day(s), week(s), month(es), year(s), hour(s), minute(s), second(s) con + (añadido a la fecha) o - o ago (resta a la fecha especificada). Si no se especifica la fecha, será la fecha en curso.

Dentro de 10 días:

```
date --date "10 days"
lun junio 30 13:13:05 CEST 2014
```

Mañana:

```
date --date "tomorrow"
```

Ayer:

```
date --date "yesterday"
```

Una semana después de la Navidad de 2014:

```
date --date "12/25/2014 23:59:00 + 1 week"
jue ene  1 23:59:00 CET 2015
```

b. hwclock

El comando **hwclock** permite interrogar directamente al reloj hardware RTC. El parámetro `--show`(por defecto) visualiza la fecha actual. Es diferente del tiempo del sistema que proviene de ntp o fecha.

```
# hwclock --show
jue 10 julio 2014 13:16:14 CEST  -0.390436 segundos
```

Al final de la visualización se proporciona el tiempo de ejecución del comando.

```
# time hwclock --show
vie 12 dic 2014 19:43:04 CEST  -0.921798 segundos

real    0m0.923s
user    0m0.000s
sys    0m0.000s
```

No es posible formatear el resultado del comando.

2. Modificar el reloj físico

Sólo se puede modificar el reloj físico como root mediante los comandos **date** (reloj del sistema interno al núcleo) y **hwclock** (reloj físico).

a. Mediante date

Modifique la fecha y la hora con el parámetro `-s`:

```
# date -s "12/12/2014 12:41:00"
```

```
vie dic 12 12:41:00 CET 2014
# date
vie dic 12 12:41:03 CET 2014
```

b. Mediante hwclock

El comando **hwclock** modifica el reloj físico (RTC) o el reloj del sistema. Como el reloj material es independiente de la hora del sistema, los resultados pueden sorprender.

```
# hwclock --set --date "12/12/2014 14:00"
# date
vie dic  12 14:00:00 CET 2014
# hwclock
vie 12 dic 2014 14:00:13 CET  -0.658230 segundos
```

Puede sincronizar la hora del sistema y la hora física en los dos sentidos. Para que se sincronice la hora física desde la hora del sistema:

```
# hwclock --systohc
# date
vie  dic  12    14:04:03   CET  2014
```

Para realizar lo contrario:

```
# hwclock --hctosys
```

3. NTP

a. Fundamentos

NTP (*Network Time Protocol*) es un protocolo que permite sincronizar los relojes de los ordenadores mediante la red y, en particular, TCP/IP, o sea Internet. Como nuestros ordenadores utilizan relojes de cuarzo, a veces éstos se pueden adelantar o atrasar mucho dependiendo de la calidad de los componentes.

Existen numerosos ámbitos en los que no se puede admitir que un sistema no esté a la hora, en particular por razones de sincronización muy precisas.

Un servidor NTP emite la hora en formato UTC. El cliente recupera la hora y la adapta en función de su huso horario. El servidor no gestiona tampoco los cambios de hora.

Si el servidor NTP está al día, la hora es muy precisa. Se codifica en 64 bits:

- los primeros 32 bits dan el número de segundos desde el 1º de enero de 1900 a medianoche (de manera que el bug NTP tendrá lugar antes del bug Unix);
- los últimos 32 bits dan la precisión de los segundos.

El nuevo protocolo NTP4 da una precisión de los segundos en 64 bits, lo que evita un inoportuno error futuro.

Encontrará en la URL siguiente una lista de algunos servidores NTP españoles:http://www.pool.ntp.org/es/zone/es

b. Cliente ntp

El servicio ntpd permite sincronizar una máquina junto a un servidor de tiempo.

```
$ ps -ef|grep nt[p]
```

```
ntp       1180     1  0 19:56 ?        00:00:00 /usr/sbin/ntpd -u ntp:ntp -g
```

El archivo de configuración es **/etc/ntpd.conf** o **/etc/ntp.conf**. En principio, este archivo ya contiene un determinado número de líneas que es preferible no tocar. Usted puede, incluso debe, añadir una línea que apunta al servidor de tiempo que ha elegido (por ejemplo es.pool.ntp.org):

```
server es.pool.ntp.org
```

Vuelva a iniciar el servicio. Su máquina debe ponerse en hora.

Es posible forzar una sincronización manual con el comando **ntpdate**. Este comando utiliza como parámetro un nombre de servidor ntp.

```
# ntpdate es.pool.ntp.org
12 Dic 14:09:21 ntpdate[6551]: adjust time server 193.54.82.20
offset 0.154057 sec
```

Si no desea utilizar el servicio ntpd, puede colocar este comando en crontab todos los días o todas las horas.

Los parámetros regionales

1. i18n y l10n

Las distribuciones se instalan y funcionan en numerosos idiomas. Seguramente habrá notado que las páginas de los manuales se muestran en su idioma si se han traducido. Del mismo modo, un gran número de programas muestran su menú en el idioma en el cual se instaló o configuró el sistema.

A diferencia de numerosos editores de programas que ofrecen versiones localizadas diferenciadas de sus programas (por ejemplo, MS Office en español y en inglés son dos versiones distintas), los editores de programas libres suelen integrar directamente un soporte para numerosos idiomas o facilitan paquetes adicionales.

Hay que diferenciar entre la regionalización (en inglés, "localization") y la internacionalización:

- La **regionalización** consiste en facilitar una traducción de un producto que corresponde a la cultura local (idioma principalmente, pero también moneda o representación de los números). A título de ejemplo, si bien el castellano se utiliza en España, México, Argentina y otros países de América Latina, cada cultura dispone de sintaxis y giros que pueden cambiar, así como de monedas diferentes.
- La **internacionalización** prepara la regionalización antes, a nivel de desarrollo del programa. Un solo programa requerirá funciones de una API de internacionalización. Una función encargada de recuperar una cadena de texto en cualquier idioma dentro de una base verificará qué idioma se utiliza y sacará la cadena correspondiente. La librería de funciones`gettext` es un buen ejemplo.

En vez de utilizar un modelo fijado como éste:

```
printf("Hello");
```

un programador procederá así:

```
printf(gettext("Hello"));
```

La función `gettext` buscará la cadena correspondiente a "Hello" en los archivos de regionalización del programa en el idioma actual y sacará una versión traducida. En español, la segunda línea será:

```
Hola
```

Para ser precisos, internacionalización se suele escribir i18n (i seguido de 18 letras y luego n), y regionalización, I10n (I para localization, seguido de 10 letras y luego n).

2. Configuraciones locales

a. Herramientas de la distribución

Hay varias maneras de modificar la regionalización del sistema, incluso de un programa dado. Cada distribución brinda un módulo de configuración. Por ejemplo, openSUSE facilita dentro de YaST un módulo de gestión del idioma y lo mismo se puede hacer con Debian de la manera siguiente:

```
dpkg-reconfigure locales
```

Estos métodos sólo modifican parámetros de sistema fácilmente accesibles. Es perfectamente posible modificarlos de forma manual mediante los archivos de configuración de texto. Del mismo modo, cada entorno de escritorio facilita herramientas de regionalización mediante su centro de

configuración. Es posible, para un programa dado, pasar de uno a otro.

b. Variables de entorno

Con la consola o en el entorno de escritorio, si se deja por defecto la regionalización, se recupera la información utilizada relativa al país, el idioma o la moneda gracias a unas variables de entorno del shell. La principal se llama **LANG**. Se suele colocar esta variable en /etc/profile u otros archivos (**/etc/sysconfig/language** en openSUSE, por ejemplo). Su contenido en el puesto del autor es el siguiente:

```
LANG=es_ES.UTF-8
```

Su formato es el siguiente:

```
idioma_Pais[.norma][@variante]
```

- **idioma** es el idioma utilizado (aquí es, para España).
- **País** es el país en mayúscula (aquí ES).
- **.norma** indica la norma utilizada para la codificación de los caracteres (aquí en UTF-8).
- **@variante** precisa variante del idioma. En algunos sistemas que utilizan la codificación ISO-8859-15 se puede especificar @euro para indicar una variante que soporta el euro de esta codificación.

Para pasar al inglés norteamericano:

```
export LANG=en_US.UTF-8
```

El comando **locale** permite recuperar información sobre los elementos de regionalización soportados por su sistema:

```
seb@slyserver:/etc/sysconfig> locale -a|grep es
es_MX
es_MX@euro
es_MX.utf8
es_AR
es_AR.utf8
fr_CL
fr_CL.utf8
es_ES
es_ES@euro
es_ES.utf8
fr_PE
fr_PE@euro
fr_PE.utf8
```

Los elementos de regionalización soportados por el sistema están en **/usr/share/locale**.

La variable LANG sustituye por defecto a las demás variables de entorno de regionalización, excepto si existen a un nivel inferior. En numerosos Unix, entre los cuales se halla Linux, existe un gran número de variables que empiezan con LC_. Si LANG es la única variable, sustituye a todas estas variables.

Si llama al comando **locale** sin parámetro, muestra lo siguiente:

```
seb@slyserver:/etc/sysconfig> locale

LANG=es_ES.UTF-8
LC_CTYPE="es_ES.UTF-8"
LC_NUMERIC="es_ES.UTF-8"
```

```
LC_TIME="es_ES.UTF-8"
LC_COLLATE="es_ES.UTF-8"
LC_MONETARY="es_ES.UTF-8"
LC_MESSAGES="es_ES.UTF-8"
LC_PAPER="es_ES.UTF-8"
LC_NAME="es_ES.UTF-8"
LC_ADDRESS="es_ES.UTF-8"
LC_TELEPHONE="es_ES.UTF-8"
LC_ALL=
```

Se puede modificar y adaptar cada una de las variables LC. Veamos su significado:

- LC_CTYPE: clase de caracteres y conversión.
- LC_NUMERIC: formato numérico por defecto, diferente del de la moneda.
- LC_TIME: formato por defecto de la fecha y la hora.
- LC_COLLATE: reglas de comparación y de ordenación (por ejemplo, los caracteres acentuados).
- LC_MONETARY: formato monetario.
- LC_MESSAGES: formato de los mensajes informativos, interactivos y de diagnóstico.
- LC_PAPER: formato de papel por defecto (por ejemplo, A4).
- LC_NAME: formato del nombre de una persona.
- LC_ADDRESS: igual para una dirección.
- LC_TELEPHONE: igual para el teléfono.
- LC_ALL: reglas para todas las demás variables LC.

Veamos un ejemplo con una modificación de LC_TIME:

```
$ date
vie. dic  12 13:24:36 CET 2014
$ LC_TIME=en_US date
Fri Dec  12 13:24:52 CET 2014
```

En algunos casos, es necesario exportar la variable LANG=C dentro de un script, por ejemplo. Eso tiene como efecto restablecer la regionalización de los comandos, en general en inglés, o de tipo POSIX, con el fin de asegurar un funcionamiento óptimo de los comandos correspondientes.

c. Husos horarios

El huso horario determina el desplazamiento temporal en comparación con la hora universal UTC, permite también gestionar el cambio de hora en primavera y otoño, de forma automática.

Para reconfigurar el huso horario, o **timezone**, emplee los siguientes comandos:

- Ubuntu y Debian: **dpkg-reconfigure tzdata**
- Fedora, CentOS y RHEL: **system-config-date** o para las anteriores versiones **hedhat-config-date**
- Para los otros: **tzselect**

Los comandos **tzselect** y **tzconfig** están descontinuados en las distribuciones recientes. En este caso emplee el método manual explicado a continuación, o bien los comandos propios de su distribución.

El huso horario se determina bajo la forma Continente/ciudad. Esta información se ubica en el archivo **/etc/timezone**.

El archivo **/etc/timezone**, en formato binario, contiene la información vinculada al huso horario, como el desfase horario con respecto a la hora UTC y las reglas para el cambio de hora en verano o en invierno.

Los archivos de configuración de husos horarios se ubican en **/usr/share/zoneinfo**. Asi, /usr/share/zoneinfo/Europe/Madrid es copiado en /etc/localtime cuando se modifica el uso horario.

Si existe, la variable **TZ** contiene el huso horario.

```
$ echo $TZ
Europe/Madrid
```

3. Codificación de los caracteres

No todo el mundo utiliza el alfabeto occidental. E incluso en los países occidentales, existen numerosas variantes de un mismo alfabeto. Los ingleses no utilizan acentos, los alemanes tienen caracteres adicionales y otros países emplean más signos auxiliares. Es aún más complicado con los países que no utilizan el mismo alfabeto o que usan ideogramas.

Los nombres de los archivos pueden ser ilegibles en otro país. El texto de un archivo, también.

Cada variante necesita la aplicación de una variante para la codificación de los caracteres. Durante mucho tiempo, la tabla se conocía como ASCII. Se trata de una tabla de caracteres codificada en 1 byte. Por lo tanto, esta tabla puede contener 256 caracteres. Como los primeros 127 son invariables, contiene todos los caracteres americanos, o sea, ningún carácter acentuado, así como los caracteres de control y formateo (supresión, retorno de carro, etc.). Los siguientes 128 son libres. En España se colocan caracteres semigráficos y los acentos, y lo mismo en otros países. Esto significa que, para cada país, hay que cambiar la tabla ASCII. Además, si accede a una palabra o un nombre de archivo creado en un país y que contiene caracteres propios de este país, hay un riesgo importante de que los nombres ya no se correspondan.

Estas tablas se han normalizado con las normas ISO. Por ejemplo, para España, la tabla ISO 8859-1 corresponde a los caracteres de Europa Occidental en Unix; ISO 8859-15 después del paso al euro. En Windows es la tabla Windows-1252; en DOS, CP850, etc. Cada tabla es incompatible con las demás. Se deben adaptar las fuentes de caracteres para cada tabla. Pero lo peor es que los 256 caracteres no bastan para codificar todos los ideogramas.

Para remediar este problema, existe una norma informática llamada **unicode** compatible con el estándar ISO 10646: da a cualquier carácter de cualquier sistema de escritura de idioma, un nombre y un identificador numérico, y eso de manera unificada, sea cual sea la plataforma informática o el programa (definición inspirada de Wikipedia). Si un sistema acepta unicode y dispone de las fuentes de caracteres asociadas, entonces se visualizan los textos y los nombres de archivos correctamente, sea cual sea el idioma y los caracteres utilizados. Se podrá visualizar correctamente un nombre de archivo en chino en un sistema en francés.

Linux, así como numerosas páginas de Internet, utilizan el formato unicode UTF-8 por defecto. Actualmente, los sistemas más corrientes gestionan correctamente este formato.

Es posible convertir un archivo codificado en una tabla dada hacia otra tabla con el programa **iconv**. El parámetro `-l` le da todas las tablas soportadas. La salida está truncada, ya que ocuparía 4 páginas. La versión 2.18 soporta 1174 codificaciones diferentes.

```
slyserver:/usr/share/zoneinfo # iconv -l
...
 437, 500, 500V1, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863,
864, 865,  866, 866NAV, 869, 874, 904, 1026, 1046, 1047, 8859_1,
8859_2, 8859_3, 8859_4,  8859_5, 8859_6, 8859_7, 8859_8, 8859_9,
10646-1:1993, 10646-1:1993/UCS4,
...
```

```
 ISO646-ES2, ISO646-FI, ISO646-FR, ISO646-FR1, ISO646-GB, ISO646-HU,
ISO646-IT, ISO646-JP-OCR-B, ISO646-JP, ISO646-KR, ISO646-NO,ISO646-NO2,
...
 WINDOWS-31J, WINDOWS-874, WINDOWS-936, WINDOWS-1250, WINDOWS-1251,
WINDOWS-1252, WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256,
WINDOWS-1257, WINDOWS-1258, WINSAMI2, WS2, YU
```

Para convertir un archivo, utilice la sintaxis siguiente:

```
iconv -f WINDOWS-1252 -t UTF8 nombre_archivo
```

TCP/IP

1. Fundamentos

El origen de **TCP/IP** se sitúa en las investigaciones de la **DARPA** (*Defense Advanced Research Project Agency*), que empezaron en 1970 y desembocaron en **ARPANET**. En realidad, la DARPA financió a la universidad de Berkeley, que integró los protocolos básicos de TCP/IP dentro de su sistema **UNIX BSD 4**.

TCP/IP se popularizó gracias a su interfaz genérica de programación de intercambios de datos entre las máquinas de una red, las primitivas **sockets**, y la integración de protocolos de aplicación. El **IAB**(*Internet Activities Board*) supervisa los protocolos de TCP/IP y a otros dos organismos:

- La **IRTF** (*Internet Reseach Task Force*), responsable del desarrollo de los protocolos.
- La **IETF** (*Internet Engineering Task Force*), responsable de la red Internet.

El **NIC** (*Network Information Center*) y en España el **ESNIC** distribuyen las direcciones red. Se describe el conjunto de los protocolos de TCP/IP en los documentos **RFC** (*Request For Comments*) (ver el RFC 793).

- La capa inferior es **IP** (*Internet Protocol*).
- La capa de transporte es **TCP** (*Transmission Control Protocol*) o **UDP** (*User Datagram Protocol*).
- Las capas superiores son las capas de los protocolos de aplicación; por ejemplo:
  - **NFS** (*Network File System*): comparte los archivos a distancia.
  - **DNS** (*Domain Name System*): asociación anfitrión<->IP.
  - **FTP** (*File Transfer Protocol*): transferencia de archivos.
  - **TELNET**: emulación de un terminal de tipo texto...

La versión del protocolo IP representado es la V4. El futuro, ya presente, es el protocolo IPV6. Compatible con IPV4, propone un sistema de direcciones en 128 bits (16 bytes) que permite extender las capacidades de la red, en particular en cuestión de tamaño y direccionamiento.

2. Direccionamiento

a. Clases

Es importante saber antes de la instalación a qué tipo de red se debe integrar el nuevo servidor, TCP/IP por supuesto, pero hay que reservarle una dirección IP, un hostname (nombre de máquina de red), conocer las diversas pasarelas, el nombre de dominio, la clase utilizada y la máscara de subred o netmask.

Hagamos un breve repaso de las clases de IP. Se define una dirección IP en 32 bits y se representa con cuatro nombres separados por puntos: **n1.n2.n3.n4**. Esta dirección está formada por dos partes que definen la dirección de red y el anfitrión en la red, respectivamente.

Según los casos, se distinguen cuatro o cinco clases de direcciones: A, B, C, D y E, pero sólo las tres primeras nos interesan.

Leyenda: N y h son bits; N es el identificador de la red h, que es el identificador de la máquina.

```
Clase A: 0NNNNNNN hhhhhhhh hhhhhhhh hhhhhhhh o sea 1.x.x.x a
126.x.x.x.
n1 está comprendido entre 1 y 126.
```

```
16777214 anfitriones, 127 redes.

Clase B: 10NNNNNN NNNNNNNN hhhhhhhh hhhhhhhh o sea de 128.0.x.x a
191.255.x.x.
n1 está comprendido entre 128 y 191.
65534 anfitriones, 16382 redes.

Clase C: 110NNNNN NNNNNNNN NNNNNNNN hhhhhhhh o sea de 192.0.0.x a
223.255.255.x.
n1 está comprendido entre 192 y 223.
254 anfitriones, 2097150 redes.

Clase D: Empieza por 1110, para la multidifusión IP.
Clase E: Empieza por 1111 para experimentación.
```

Existen direcciones de anfitriones que no pueden utilizarse. Por ejemplo, en la clase C sólo se pueden tener 254 anfitriones, mientras que el identificador de la máquina está codificado en 8 bits (por lo tanto, 256 valores). La dirección 0 representa la dirección de la red, y la dirección 255, la del**broadcast** (multidifusión).

Observe que las direcciones siguientes no se deben enrutar en Internet y se reservan a redes locales.

- 10.0.0.0 - 10.255.255.255 (10/8)
- 172.16.0.0 - 172.31.255.255 (172.16/12)
- 192.168.0.0 - 192.168.255.255 (192.168/16)

La dirección 127.0.0.1 es la dirección de loopback o bucle: representa a la propia máquina, así como la subred 127.0.0.0/8.

b. Subredes

Es posible dividir estas redes en subredes usando máscaras que permiten una división más pequeña de las direcciones. Un **netmask** es una máscara binaria que permite separar de manera inmediata la dirección de la red y de la subred, de la dirección del anfitrión en la dirección IP global. Las máscaras predefinidas son:

- **Clase A**: 255.0.0.0
- **Clase B**: 255.255.0.0
- **Clase C**: 255.255.255.0

Para comunicarse directamente entre ellos, los anfitriones deben pertenecer a una misma red o subred. Calcular una subred es bastante sencillo. Veamos un ejemplo para una red de clase C.

- Red: 192.168.1.0
- Dirección de red: 192.168.1.255
- Máscara de red: 255.255.255.0

Calcular una máscara de subred:

- Para calcular la máscara de subred, primero debe determinar cuántas máquinas quiere integrar en ella. Una red de clase C permite integrar 254 máquinas (0 y 255 están reservados). Desea crear redes que contienen 60 máquinas. Añada 2 a este valor para las direcciones reservadas (dirección de la subred y dirección de broadcast), lo que da **62**.
- Una vez determinado el número de máquinas, encuentre la potencia de dos, exacta o superior al número encontrado. 2 elevado a 6 da **64**.
- Escriba la máscara en binario, coloque todos los bits de la máscara de red de clase C a 1, y coloque a 0 los 6 primeros bits de la máscara correspondiente a la parte máquina: **11111111**

11111111 11111111 11000000

- Convierta esta máscara en decimal: **255.255.255.192**, y calcule el conjunto de las subredes posibles. Como está en una red de clase C, todavía puede hacer variar los dos últimos bits de la parte máquina:
  - 00xxxxxx: 255.255.255.0
  - 01xxxxxx: 255.255.255.64
  - 10xxxxxx: 255.255.255.128
  - 11xxxxxx: 255.255.255.192
- Al final, obtiene cuatro subredes de 62 máquinas, o sea, 248 máquinas. En efecto: obtiene 256 si añade las cuatro direcciones de broadcast y las cuatro direcciones de red.

c. Encaminamiento

La máscara de red permite determinar si una máquina destinataria está en la misma red que usted o no. Hay que indicar la ruta que deben tomar los paquetes IP para alcanzar su destino. Si su máquina es un terminal cliente que dispone de una sola tarjeta de red y esta red tiene un único enrutador (caso clásico de una conexión hacia Internet), entonces debe crear dos rutas. La primera es la que indica qué tarjeta de red deben emplear los paquetes para acceder al resto de la red (a la subred); la segunda, qué ruta deben utilizar los paquetes para salir de la red. En general, se habla de ruta por defecto cuando sólo hay un enrutador.

- Hacia red1 -> utilizar interfaz de red izquierda.
- Hacia red2 -> utilizar interfaz de red derecha.
- Hacia otras -> utilizar interfaz de red derecha hacia enrutador1.

Ejemplo:

Red1 de clase C 192.168.1.0 en eth0, Red2 de clase B 172.16.0.0 en eth1, dirección de enrutador 192.168.1.254.

Red	Máscara	Interfaz	Pasarela
192.168.1.0	255.255.255.0	eth0	eth0
172.16.0.0	255.255.0.0	eth1	eth1
0.0.0.0	0.0.0.0	eth0	192.168.1.254

Todos los paquetes de red hacia 192.168.1.0 transitarán por eth0. Todos los paquetes cuyo destino sea 172.16.0.0 transitarán por eth1. Por defecto, los demás paquetes para las redes no especificadas transitarán por eth0 y serán tratados por la pasarela 192.168.1.254, que enrutará los paquetes.

d. IPv6

Saturación de IPv4

Las últimas direcciones IPv4 públicas se saturaron el 3 de febrero de 2011, aunque no se usen todas debido al modo de reparto por clases y subredes. Por ejemplo, las únicas subredes de clase A representan cerca de dos mil millones de direcciones IP, probablemente sin usarse todas o en manos de organismos que no usan todas. Incluso sucede lo mismo con las subredes de clase B, con mil millones de direcciones.

Un gran número de direcciones IP se reservan para usos concretos, como por ejemplo las IP multicast, que no pueden asignarse a un equipo.

Una mala gestión inicial, la multiplicación de los terminales, especialmente de terminales móviles de tipología diversa y el gran crecimiento del número de conexiones a Internet han incrementado la

demanda de forma espectacular, lo que nos ha llevado a su extinción.

Se han aplicado varios métodos para intentar retrasar la fecha de saturación, especialmente haciendo desaparecer el concepto de clases en las IP públicas o el uso de NAT, que permite mediante la traducción de direcciones ofrecer servicios accesibles al público alojados en máquinas pertenecientes a una subred privada.De este modo puede instalar servidores web u otros servicios en sus propios PC de casa, que tienen una dirección IP de una red privada, pero son accesibles desde el exterior mediante la IP pública y las funciones NAT de su router de Internet.

Las direcciones IPv4 no usadas que todavía están en manos de sus propietarios originales ahora son objeto de un negocio lucrativo.

Esto no significa la inmediata desaparición de IPv4: estas direcciones IP se seguirán usando durante muchos años, lo que tarden las IP no utilizadas en dejar de estarlo, lo que se debería producir entre 2014 y 2015. Las direcciones IPv4 públicas están llamadas a desaparecer o a cohabitar con otro mecanismo.

Direccionamiento IPv6

Las direcciones IPv6 reemplazarán a las direcciones IPv4. Se codifican con 128 bits y hay tantas disponibles que cada ser humano podría disponer de un trillón de direcciones sin llegar a saturarse. De este modo, NAT se convierte en una tecnología inútil y el encaminamiento se vuelve más sencillo.

La dirección se divide en 8 grupos de dos bytes escritos en hexadecimal y separados por ":", es decir 39 caracteres:

```
2001:0e36:2ed9:d4f0:021b:0000:0000:f81d
```

Pueden omitirse de uno a tres ceros no significativos:

```
2001:e36:2ed9:d4f0:21b:0:0:f81d
```

Uno o varios bloques de 16 bytes (2 bytes) nulos pueden omitirse conservando los ":" de cada lado. En el ejemplo usado anteriormente, sólo se puede aplicar una solución.

```
2001:e36:2ed9:df40:21b:0:0:f81d
```

El prefijo predeterminado para las direcciones es /64. Lo que significa que su proveedor de Internet le proporcionará una dirección con el formato 2a01:e36:2ed9:d4f0::/64, es decir de forma desarrollada:

```
2001:0e36:2ed9:d4f0:0000:0000:0000:0000/64
```

Esta subred es suya y se podrá direccionar automáticamente por su proveedor de Internet. No se puede (todavía) subdividirla en otras subredes. Podrá crear tantas direcciones IPv6 como desee en esta subred, alrededor de 2^{64} huéspedes: 18.446.744.073.709.551.616 direcciones IP. Se podrá acceder directamente a estas IP desde el exterior, sin NAT, pero evidentemente podrá proteger sus equipos con un firewall.

Uso en Linux

Este libro se basa en la configuración de las interfaces y de los servicios para un uso en IPv4. Sin embargo, el principio es el mismo que en IPv6. Linux es completamente compatible con IPv6, de forma nativa. Los comandos ifconfig, route, ping, etc. aceptan IPv6, así como los archivos de configuración de las interfaces propias de cada distribución. A continuación se muestran algunos ejemplos adaptados.

3. Configuración

a. El caso de las distribuciones de tipo Red Hat/Fedora

Red Hat propone herramientas para configurar la red básica sin tener que manejar los archivos de configuración. El programa en modo de texto **netconfig** permite instalar la configuración TCP/IP básica (IP estática o dinámica, enrutador, nombre de anfitrión, servidor de nombres).

```
# netconfig --device eth0
```

El comando gráfico **system-config-network** inicia una interfaz más completa.

Las distribuciones Mandriva y openSUSE retoman el mismo principio que el que se expone a continuación, pero la sintaxis y la posición de los archivos pueden variar. Vea la documentación de su distribución para más detalles.

Interfaces de red

La configuración básica de una interfaz de red se hace con la ayuda del comando **ifconfig**. Sin embargo, la mayoría de las distribuciones utilizan scripts de administración y archivos de configuración que simplifican mucho las cosas, ya que configuran al mismo tiempo la interfaz de red y las rutas.

Configuración de eth0 para la dirección de clase C 192.168.1.2

```
ifconfig eth0 inet 192.168.1.2 netmask 255.255.255.0
ifconfig eth0 192.168.1.2
```

Activación de la interfaz de red eth0:

```
ifconfig eth0 up
```

Parada de la interfaz de red eth0:

```
ifconfig eth0 down
```

Visualización de información de eth0:

```
# ifconfig eth0
eth0      Vínculo encap:Ethernet  HWaddr 00:XX:XX:XX:XX:XX
          inet adr:192.168.1.60  Bcast:192.168.1.255  Máscara:255.255.255.0
          adr inet6: fe80::21b:fcff:fec9:f81d/64 Scope:Vínculo
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16522 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13631 errors:0 dropped:0 overruns:0 carrier:2
          collisions:0 lg file transmission:1000
          RX bytes:17732221 (16.9 MB)  TX bytes:1648879 (1.5 MB)
```

Observe que la dirección IPv6 aparece en el resultado en la línea inet6.

Configuración de una dirección IPv6

```
# ifconfig eth0 inet6 add fe80::21b:fcff:fec9:f81d/64
```

Visualización de todas las interfaces de red activadas:

```
ifconfig
```

Visualización de todas las interfaces de red, activadas o no:

```
ifconfig -a
```

Lo mejor sigue siendo la utilización de los scripts **ifup** e **ifdown**. Se basan en los archivos presentes en **/etc/sysconfig/network-scripts/**. Estos archivos de configuración de interfaz se llaman ifcfg-xxx, donde xxx es el nombre de la interfaz de red, como eth0.

```
DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=static
```

Los parámetros hablan por sí mismos. Los valores **NETWORK** y **BROADCAST** son opcionales si se notifican **IPADDR** y **NETMASK** (en este caso, el cálculo es automático) o si se utiliza **DHCP**.**BOOTPROTO** indica cómo montar la interfaz, o bien **static**, o bien **dhcp**. Se puede utilizar el valor**bootp**. En el caso de DHCP, el archivo se puede parecer a esto:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

En el caso de una configuración estática, **IPADDR** y **NETMASK** son obligatorios:

```
DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=static
```

ONBOOT determina si se debe activar automáticamente la interfaz en el arranque de la máquina.

Para IPv6, debe utilizar las siguientes variables:

```
IPV6INIT=yes
IPV6ADDR=<IPv6-IP-Address>
IPV6_DEFAULTGW=<IPv6-IP-Gateway-Address>
```

IPV6INIT indica si IPv6 está activo o no en esta interfaz. Las dos variables siguientes indican la dirección IP y su puerta de enlace.

Una vez notificado correctamente el archivo, se utilizan los comandos **ifup/ifdown**:

Activación de la interfaz eth0:

```
ifup eth0
```

Parada de la interfaz eth0:

```
ifdown eth0
```

Parámetros generales

El archivo **/etc/sysconfig/network** contiene los parámetros generales de la red.

```
NETWORKING=yes
HOSTNAME=puesto1.mired.org # nombre completo
GATEWAY=0.0.0.0 # pasarela por defecto
NISDOMAIN= # nombre del dominio NIS
NETWORKING_IPV6=yes
```

NETWORKING: activación o no de la red.

- **HOSTNAME**: nombre de dominio completo FQDN.
- **GATEWAY**: dirección IP de la pasarela.
- **GATEWAYDEV**: interfaz de red que permite acceder a la pasarela.
- **NISDOMAIN**: en caso de un dominio NIS.
- **NETWORKING_IPV6**: activación o no del soporte de IPv6.

b. Máquinas de tipo Debian

El archivo de configuración de las interfaces de redes en Debian (y Ubuntu) se sitúa en **`/etc/network/interfaces`**. No tiene el mismo formato que en Red Hat.

```
# cat interfaces
auto lo eth0 eth1
iface lo inet loopback

iface eth0 inet static
        address 192.161.1.60
        netmask 255.255.255.0
        broadcast 192.168.1.255
        gateway 192.168.1.1

iface eth1 inet dhcp
```

Este ejemplo muestra tres tipos de interfaces:

- la interfaz lo de loopback,
- la interfaz eth1 en dhcp, que no necesita una configuración más avanzada,
- la interfaz eth0 configurada de manera estática.

La sintaxis general de una declaración es la siguiente:

```
interfaz nombre tipo modo
```

Con una configuración estática, especifique los diferentes parámetros con las palabras claves siguientes:

- **address**: la dirección IP.
- **netmask**: la máscara de subred.
- **broadcast**: la dirección de broadcast.
- **gateway**: la pasarela por defecto.

La línea **auto** indica las interfaces que se activarán automáticamente en el arranque.

Para IPv6 la configuración es idéntica: basta con reemplazar inet con inet6.

A partir de 2012 podemos emplear una directiva **include** que permite incluir un archivo externo a la configuración.

El archivo **`/etc/hostname`** contiene el nombre de la máquina:

```
# cat /etc/hostname
slyserver
```

c. Encaminamiento

Con la utilización de los archivos y comandos anteriores, no hace falta crear un encaminamiento específico, puesto que la pasarela por defecto está ya presente (vea parámetros generales) e **ifup**implementa automáticamente las rutas para las interfaces. Sin embargo, se puede utilizar el comando **route**.

Visualiza las rutas actuales:

```
route
netstat -nr
```

En el ejemplo siguiente, las interfaces vmnet1 y vmnet8 son interfaces virtuales procedentes de la configuración de red de VMWare.

```
# netstat -rn
Tabla de encaminamiento IP del núcleo
Destino         Pasarela      Genmask         Indic   MSS Ventana irtt
Iface
192.168.211.0   0.0.0.0       255.255.255.0   U         0 0          0
vmnet8
192.168.1.0     0.0.0.0       255.255.255.0   U         0 0          0
eth0
172.16.248.0    0.0.0.0       255.255.255.0   U         0 0          0
vmnet1
169.254.0.0     0.0.0.0       255.255.0.0     U         0 0          0
eth0
127.0.0.0       0.0.0.0       255.0.0.0       U         0 0          0
lo
0.0.0.0         192.168.1.1   0.0.0.0         UG        0 0          0
eth0
```

Creación de la entrada loopback:

```
route add -net 127.0.0.0
```

Crea la ruta hacia la red 192.168.1.0, que pasa por eth0. Se puede omitir netmask.

```
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

Crea la pasarela por defecto hacia el enrutador:

```
route add default gw 192.168.1.254
```

Suprime la ruta hacia la red 172.16.0.0:

```
route del -net 172.16.0.0 eth0
```

La gestión de rutas IPv6 es idéntica, pero indicando el tipo de dirección usado en el comando con la opción -A:

```
route -A inet6 add <ipv6> gw <ipv6>
```

4. Herramientas de red

a. FTP

Resulta útil conocer el comando **ftp** (*file transfer protocol*). Permite la transferencia de archivos entre dos máquinas. El parámetro que adopta es el nombre de la máquina remota. Para que el comando**ftp** funcione, el servicio ftp debe funcionar en la máquina remota y en el puerto 21.

Veamos un ejemplo (poco práctico) de conexión con error y nuevo intento.

```
ftp> open
(to) machine
Connected to machine.
220 machine FTP server (Digital UNIX Version 5.60) ready.
Name (machine:root): root
331 Password required for root.
Password:
530 Login incorrect.
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> user
(username) root
331 Password required for root.
Password:
230 User root logged in.
ftp> pwd
257 "/" is current directory.
```

Sin embargo, lo más sencillo es:

```
$ ftp machine
Connected to machine.
220 machine FTP server (Digital UNIX Version 5.60) ready.
Name (machine:root): root
331 Password required for root.
Password:
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Veamos una lista de comandos **ftp**.

Comando	Acción
open	Seguido de un nombre de máquina, abre una conexión en la máquina especificada.
user	Inserción del usuario remoto para una conexión.
quit	Fin de la conexión y del comando ftp.
ascii	Transferencia de los archivos en modo ASCII (conversión de los caracteres especiales y fin de línea en MS y Unix, por ejemplo).
binary	Transferencia de los archivos en modo binario.
glob	Suprime la interpretación de los caracteres especiales.
help	Muestra la ayuda.
prompt	Seguido de on u off, activa o desactiva la confirmación individual de transferencia para cada archivo (mget o mput).
pwd	Muestra el directorio remoto actual.
cd	Seguido de la ruta, desplazamiento en el árbol remoto.
ls	Lista los archivos de la máquina remota.
delete	Seguido de un nombre de archivo, suprime el archivo remoto.
mdelete	Múltiple. Suprime los archivos remotos.

get	Recupera el archivo remoto.
mget	Múltiple. Recupera los archivos remotos (lista o modelo).
put	Manda el archivo local hacia la máquina remota.
mput	Múltiple. Manda los archivos locales a la máquina remota (lista o modelo).
close/ disconnect	Cierra la sesión actual.
lcd	Cambia de directorio en la máquina local.
hash	Durante la transferencia, escribe una "#" en pantalla para cada buffer transferido.
system	Información sobre el sistema remoto.
recv	Recepción de un archivo.
send	Envío de un archivo.
rename	Renombra un archivo remoto.
mkdir	Crea un directorio en la máquina remota.
rmdir	Suprime un directorio en la máquina remota.
!comando	Ejecuta el comando local.

b. Telnet

Telnet es un cliente ligero que permite abrir una conexión y una sesión en una máquina remota que ofrece un servidor telnet o cualquier otro servicio que permita la comunicación en texto "en claro": smtp, http, etc. A menudo, se inicia este servidor desde xinetd o inetd. Por tanto, Telnet es ideal para verificar el funcionamiento de varios protocolos. Su sintaxis es muy simple:

```
$ telnet -l user machine port
Ejemplo:
# telnet 192.168.1.60
Trying 192.168.1.60...
Connected to 192.168.1.60.
Escape character is '^]'.
Welcome to openSUSE 10.3 (i586) - Kernel 2.6.24.4-default (3).

slyserver login: seb
Contraseña:
Última conexión: jueves 15 de mayo de 2008 a 06:26:30 CEST de consola
en:0
Tiene un nuevo mensaje.
Have a lot of fun...
seb@slyserver:~>
```

¡Cuidado! El servicio (y el cliente) telnet no tienen ningún tipo de seguridad: las conexiones transitan sin cifrar por la red, y cualquier esnifador IP (wireshark, tcdump por ejemplo) puede interceptar y ver todo lo que se hace. Incluso se transmite la contraseña sin cifrar. Evite utilizar este servicio y céntrese en OpenSSH.

c. Ping

El comando **ping** es un comando central, incluso ineludible. Para saber si una máquina está accesible o no, lo primero que se suele hacer es intentar "hacer ping" (con la condición de que la configuración del firewall autorice las peticiones ICMP). El comando **ping6** hace lo mismo para una dirección Ipv6. Si se omiten los paquetes, es señal de un error en la red. Si se omiten todos los paquetes, será que la conexión el equipo remoto está cortada, o bien el protocolo ICMP está

restringido por un firewall.

Ping emite un "eco" de red, como un sónar, y espera una respuesta, el retorno del eco. Para ello, utiliza el protocolo ICMP. Interrumpa el comando **ping** con [Ctrl] **C**.

```
$ ping www.kde.org
PING www.kde.org (62.70.27.118) 56(84) bytes of data.
64 bytes from 62.70.27.118: icmp_seq=1 ttl=57 time=10.5 ms
64 bytes from 62.70.27.118: icmp_seq=2 ttl=57 time=11.3 ms
64 bytes from 62.70.27.118: icmp_seq=3 ttl=57 time=10.4 ms
64 bytes from 62.70.27.118: icmp_seq=4 ttl=57 time=11.5 ms
...
```

Tres parámetros deben llamarle la atención:

- `-c` permite especificar el número de ecos que se deben emitir.
- `-b` permite emitir un eco en una dirección de broadcast.
- `-I` permite especificar la interfaz de red.

En el primer caso, el parámetro puede ser útil en un script para probar que un servidor responde:

```
# ping -c 1 10.9.238.170 >/dev/null 2>&1 && echo "El servidor contesta"
El servidor contesta
```

En el segundo caso, todas las direcciones de la subred correspondientes a la dirección de broadcast deben responder.

```
# ping -b 192.168.1.255
WARNING: pinging broadcast address
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.232 ms
64 bytes from 192.168.1.60: icmp_seq=1 ttl=64 time=0.240 ms
64 bytes from 192.168.1.130: icmp_seq=1 ttl=255 time=0.285 ms
64 bytes from 192.168.1.139: icmp_seq=1 ttl=255 time=0.292 ms
...
```

En el último caso, puede especificar una tarjeta de salida. Esta opción es muy útil para comprobar una resolución DNS o una ruta.

```
# ping -I eth0 192.168.1.60
PING 192.168.1.60 (192.168.1.60) from 192.168.1.10:eth0: 56(84) bytes
of data.
64 bytes from 192.168.1.60: icmp_seq=1 ttl=62 time=0.478 ms
64 bytes from 192.168.1.60: icmp_seq=2 ttl=62 time=0.408 ms
...
```

d. Traceroute

Cuando intenta acceder a un anfitrión remoto desde su máquina, los paquetes IP pasan a menudo por muchas rutas, a veces diferentes según el punto de partida y de destino, los cuellos de botella, etc. El trayecto pasa por numerosas pasarelas (gateways), que dependen de las rutas por defecto o predefinidas de cada una de ellas.

El comando **traceroute** permite visualizar cada uno de los puntos de paso de sus paquetes IP a su destino en un anfitrión dado. En el ejemplo siguiente, el anfitrión ubicado en la región parisina en la red del proveedor Free intenta determinar la ruta recorrida para ir al servidor www.kde.org. Se enmascarará la dirección IP fuente (fuera de la red local). El comando **traceroute6** es idéntico para IPv6.

```
$ traceroute www.kde.org
traceroute to www.kde.org (62.70.27.118), 30 hops max, 40 byte packets
 1 DD-WRT (192.168.1.1)  0.558 ms   0.533 ms   0.585 ms
 2 82.xxx.yyy.zzz (82.xxx.yyy.zzz)  6.339 ms   6.404 ms   6.901 ms
 3 * * *
 4 * * *
 5 212.73.205.5 (212.73.205.5)  39.267 ms   35.499 ms   31.736 ms
 6 ae-12-55.car2.Paris1.Level3.net (4.68.109.144)  6.485 ms ae-22-
52.car2.Paris1.Level3.net (4.68.109.48)  6.401 ms   6.338 ms
 7 UUnet-Level3.Level3.net (212.73.240.206)  6.113 ms   6.152 ms
5.866 ms
 8 so-3-2-0.TL2.PAR2.ALTER.NET (146.188.8.121)  6.107 ms   6.410 ms
6.365 ms
 9 so-2-2-0.TL2.STK2.ALTER.NET (146.188.7.33)  87.323 ms   86.840 ms
87.010 ms
10 so-7-1-0.XR2.OSL2.ALTER.NET (146.188.15.62)  96.491 ms   97.148 ms
96.488 ms
11 ge-0-1-0.GW6.OSL2.ALTER.NET (146.188.3.242)  95.972 ms   95.934 ms
96.108 ms
12 213.203.63.74 (213.203.63.74)  95.320 ms   94.321 ms   96.188 ms
13 leeloo.troll.no (62.70.27.10)  94.064 ms   94.052 ms   92.374 ms
14 jamaica.kde.org (62.70.27.118)  97.064 ms   96.182 ms   97.853 ms
```

e. Whois

¿Sabía que puede obtener toda la información que desee sobre un dominio (pepito.es) usando el comando **whois**? Por ejemplo, para obtener toda la información sobre el dominio kde.org:

```
> whois kde.org
...
Domain Name:KDE.ORG
Domain ID: D1479623-LROR
Creation Date: 1996-12-14T05:00:00Z
Updated Date: 2013-12-14T12:11:24Z
Registry Expiry Date: 2014-12-13T05:00:00Z
Sponsoring Registrar:GoDaddy.com, LLC (R91-LROR)
Sponsoring Registrar IANA ID: 146
WHOIS Server:
Referral URL:
Domain Status: clientDeleteProhibited
Domain Status: clientRenewProhibited
Domain Status: clientTransferProhibited
Domain Status: clientUpdateProhibited
Registrant ID:CR88351429
Registrant Name:K Desktop Environment EV
Registrant Organization:KDE e.V.
Registrant Street: Schoenhauser Allee 6
Registrant City:Berlin
Registrant State/Province:
Registrant Postal Code:10119
Registrant Country:DE
Registrant Phone:+49.30202373050
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email:domain@kde.org
Admin ID:CR88351433
...
```

f. Netstat

El comando **netstat** permite obtener gran cantidad de información sobre la red y los protocolos.

El parámetro `-i` permite obtener el estado de las tarjetas de redes para determinar una posible

avería o un problema de cable:

```
# netstat -i
Tabla de interfaces del núcleo
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP
TX-OVR Flg
eth0   1500   0 2332007      0      0      0   677842      0      0
0 BMRU
lo    16436   0    1109      0      0      0     1109      0      0
0 LRU
```

Si añade el parámetro `-e`, obtiene el mismo resultado que con ifconfig `-a`.

```
# netstat -ei
Tabla de interfaces del núcleo
eth0    Vínculo encap:Ethernet  HWaddr 00:XX:D3:XX:AA:XX
        inet adr:12.168.1.60  Bcast:192.168.1.255  Máscara:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:2335314 errors:0 dropped:0 overruns:0 frame:0
        TX packets:678095 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        RX bytes:1055212145 (1006.3 MB)  TX bytes:61264196 (58.4 MB)
        Interruption:20 Dirección basica:0x8c00

lo      Vínculo encap:Bucle local
        inet adr:127.0.0.1  Máscara:255.0.0.0
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:1109 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1109 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:0
        RX bytes:60423 (59.0 Kb)  TX bytes:60423 (59.0 KB)
```

El parámetro `-r` permite obtener las tablas de encaminamiento como ruta. Añada el parámetro `-n`para indicar las IP en lugar de los nombres.

```
# netstat -rn
Tabla de encaminamiento IP del núcleo
Destino         Pasarela      Genmask         Indic   MSS Ventana irtt
Iface
192.168.211.0   0.0.0.0       255.255.255.0   U         0 0          0
vmnet8
192.168.1.0     0.0.0.0       255.255.255.0   U         0 0          0
eth0
172.16.248.0    0.0.0.0       255.255.255.0   U         0 0          0
vmnet1
169.254.0.0     0.0.0.0       255.255.0.0     U         0 0          0
eth0
127.0.0.0       0.0.0.0       255.0.0.0       U         0 0          0
lo
0.0.0.0         192.168.1.1   0.0.0.0         UG        0 0          0
eth0
```

El parámetro `-a` permite visualizar todas las conexiones, para todos los protocolos, incluido los puertos en escucha de la máquina. La salida es demasiado larga para reproducirla en estas páginas.

```
# netstat -a | wc -l
495
```

El parámetro `-A` permite especificar el protocolo que es preciso consultar: inet, inet6 (ipv6), unix, ipx, ax25, netrom y ddp.

```
# netstat -a -A inet
Conexiones Internet activas (servidores y establecidas)
Proto Recv-Q Send-Q Dirección local         Dirección remota
Estado
tcp        0      0 localhost:716           *:*
LISTEN
tcp        0      0 *:sunrpc                *:*
LISTEN
tcp        0      0 localhost:ipp           *:*
LISTEN
tcp        0      0 localhost:smtp          *:*
LISTEN
tcp        0      0 localhost:hpssd         *:*
LISTEN
tcp        0      0 slyserver:41851         imap.tele2.es:imap
ESTABLISHED
tcp        0      0 slyserver:41850         imap.tele2.es:imap
ESTABLISHED
tcp        0      0 slyserver:54220         by1msg4176111.gate:msnp
ESTABLISHED
tcp        0      0 slyserver:34267         by2msg2105007.phx.:msnp
ESTABLISHED
tcp        0      0 slyserver:47990         by1msg4082314.phx.:msnp
ESTABLISHED
udp        0      0 *:filenet-tms           *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:ipp                   *:*
udp        0      0 172.16.248.1:ntp        *:*
udp        0      0 192.168.211.1:ntp       *:*
udp        0      0 slyserver:ntp           *:*
udp        0      0 localhost:ntp           *:*
udp        0      0 *:ntp                   *:*
raw        0      0 *:icmp                  *:*                     7
```

Y para terminar, el parámetro `-p` permite indicar, cuando es posible, el PID y el nombre del proceso.

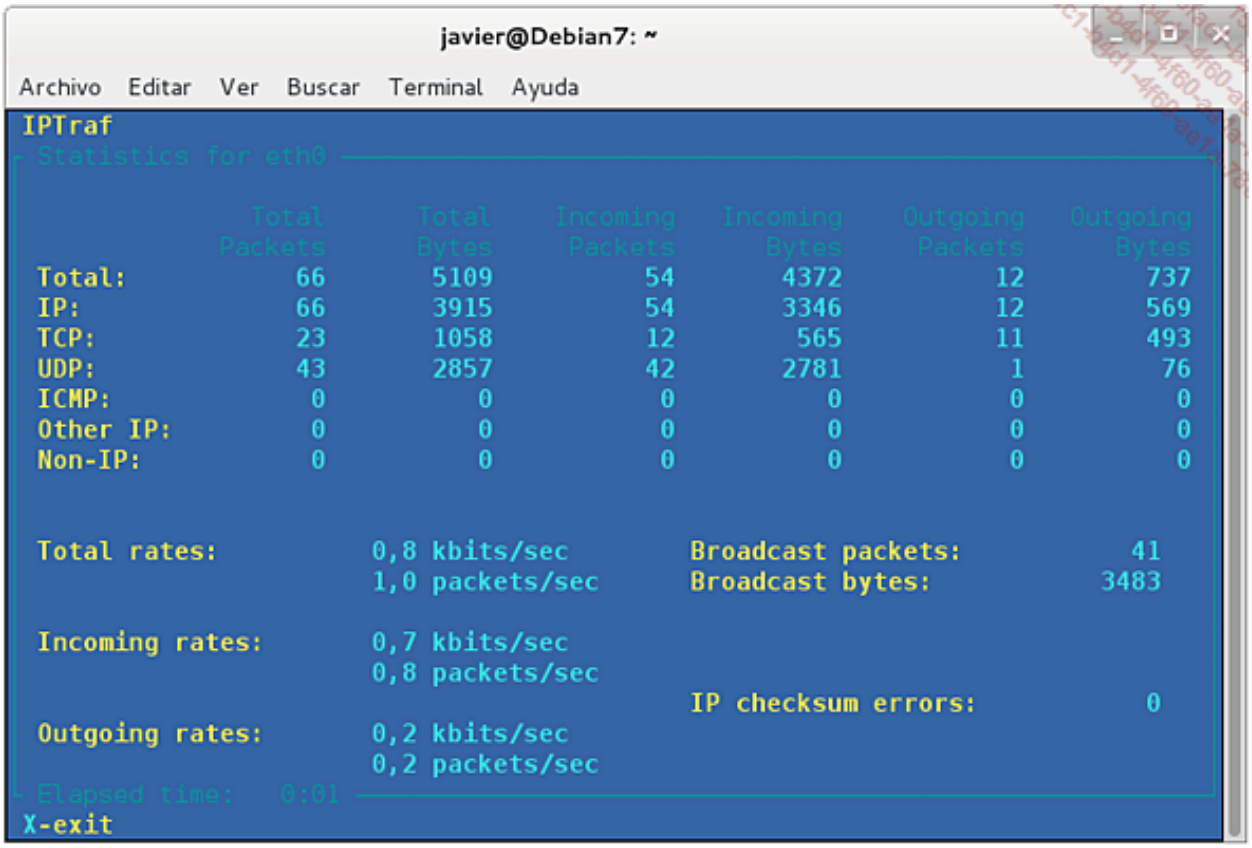
```
# netstat -A inet -p
Conexiones a Internet activas (sin servidores)
Proto Recv-Q Send-Q Dirección local          Dirección remota
Estado        PID/Program name
...
tcp        0      0 slyserver:54220          by1msg4176111.gate:msnp
ESTABLISHED   4041/kopete
tcp        0      0 slyserver:34267          by2msg2105007.phx.:msnp
ESTABLISHED   4041/kopete
tcp        0      0 slyserver:47990          by1msg4082314.phx.:msnp
ESTABLISHED   4041/kopete
```

g. IPTraf

El comando **iptraf** permite visualizar en tiempo real la actividad de la red mediante una herramienta de texto, opcionalmente interactiva (línea de comandos). Los menús son claros. Puede moverse por ellos con las teclas de dirección y los diferentes atajos especificados.

La captura siguiente muestra la visualización detallada de las estadísticas de la tarjeta `eth0`. A esta pantalla se accede desde la línea de comandos con:

```
# iptraf -d eth0
```



IPTraff analiza el tráfico de eth0

5. Archivos generales

a. /etc/resolv.conf

Se utiliza el archivo **`/etc/resolv.conf`** para indicar al sistema qué servidores de nombres y qué dominios hay que consultar para resolver las peticiones DNS clientes. Estas API, al igual que las API estándares de Linux, están incluidas en la librería (no hace falta añadir herramientas adicionales). Esta librería se llama **`resolver`**.

Al configurar DHCP, en principio se actualiza automáticamente este archivo y no se debería modificar, salvo que haya prohibido la configuración DNS en el cliente.

```
$ cat /etc/resolv.conf
domain midominio.org
search midominio.org
nameserver 192.168.1.1
nameserver 192.168.1.2
```

- **domain**: nombre del dominio local. Las peticiones se suelen reducir a unos atajos relativos al dominio local. Si no está creado, el nombre del dominio se debe determinar a partir del nombre completo del anfitrión: corresponde a la parte ubicada después del primer ".".
- **search**: lista de los dominios de búsqueda. Por defecto, durante la utilización de atajos (nombres de anfitriones cortos) el resolver inicia una búsqueda sobre el dominio definido por la línea domain, pero se puede especificar aquí una lista de dominios separados por espacios o comas.
- **nameserver**: dirección IP del servidor de nombres (el servidor DNS). Se puede colocar un máximo de tres. El resolver intenta utilizar el primero. Si fracasa (timeout), pasa al segundo,

y así sucesivamente.

- **opciones**: se pueden especificar opciones. Por ejemplo, **timeout:n**, donde n (en segundos) indica el tiempo de espera de respuesta de un servidor de nombres antes de pasar al siguiente.

b. /etc/hosts y /etc/networks

Sin siquiera utilizar un servidor de nombres, se puede establecer una correspondencia entre las direcciones IP y los nombres de las máquinas dentro del archivo **/etc/hosts**.

```
192.168.1.1  server1 www1 ftp
192.168.1.11 puesto1
192.168.1.12 puesto2
```

Puede hacer lo mismo para nombrar las redes (lo que puede ser útil para los **tcp_wrappers** o el comando **route**) en el archivo **/etc/networks**.

```
loopnet  127.0.0.0
localnet 192.168.1.0
```

c. /etc/nsswitch.conf

El archivo **/etc/nsswitch.conf** permite determinar el orden en el cual el resolver (u otros servicios) recupera su información. Las dos líneas en negrita en el ejemplo indican que, durante una petición de resolución de nombre (o red), los archivos son prioritarios. Primero se lee el archivo**/etc/hosts**; luego, si el resolver no encuentra la información, busca mediante una resolución DNS.

```
passwd: compat
group:  compat
hosts:          files dns
networks:       files dns

services:       files
protocols:      files
rpc:            files
ethers:         files
netmasks:       files
netgroup:       files nis
publickey:      files

bootparams:     files
automount:      files nis
aliases:        files
```

Puede ocurrir que determinados programas no utilicen el resolver sino directamente el DNS o el archivo **etc/hosts**, o inviertan el orden establecido en **/etc/nsswitch.conf**. En este caso, no es posible prever (ni predecir) el funcionamiento correcto de este tipo de programas...

d. /etc/services

El archivo **/etc/services** contiene la lista de los servicios de red conocidos de Unix, así como los puertos y protocolos asociados. Muchos servicios (entre los cuales se halla **xinetd**) y subsistemas, como el firewall de Linux, lo utilizan.

Este archivo es indicativo: se trata de un archivo de descripción y definición: no todos los servicios de este archivo se ejecutan obligatoriamente en su máquina (por fortuna... ¡visto el número!). Y se

puede configurar un servicio para escuchar otro puerto. En cambio, se aconseja ponerlo al final cuando se añade un servicio que no está presente en este archivo.

```
tcpmux            1/tcp    # TCP Port Service Multiplexer
tcpmux            1/udp    # TCP Port Service Multiplexer
compressnet       2/tcp    # Management Utility
compressnet       2/udp    # Management Utility
compressnet       3/tcp    # Compression Process
compressnet       3/udp    # Compression Process
rje               5/tcp    # Remote Job Entry
rje               5/udp    # Remote Job Entry
echo              7/tcp    Echo
echo              7/udp    Echo
discard           9/tcp    # Discard
discard           9/udp    # Discard
systat           11/tcp    users        # Active Users
systat           11/udp    users        # Active Users
daytime          13/tcp    # Daytime (RFC 867)
daytime          13/udp    # Daytime (RFC 867)
netstat          15/tcp    # Unassigned [was netstat]
qotd             17/tcp    quote        # Quote of the Day
qotd             17/udp    quote        # Quote of the Day
msp              18/tcp    # Message Send Protocol
msp              18/udp    # Message Send Protocol
chargen          19/tcp    # Character Generator
chargen          19/udp    # Character Generator
ftp-data         20/tcp    # File Transfer [Default Data]
ftp-data         20/udp    # File Transfer [Default Data]
ftp              21/tcp    # File Transfer [Control]
fsp              21/udp    # File Transfer [Control]
ssh              22/tcp    # SSH Remote Login Protocol
ssh              22/udp    # SSH Remote Login Protocol
telnet           23/tcp    # Telnet
telnet           23/udp    # Telnet
...
```

e. /etc/protocols

El archivo **/etc/protocols** contiene la lista de los protocolos conocidos por Unix.

```
# Assigned Internet Protocol Numbers
#
# Decimal   Keyword     Protocol                          References
# -------   -------     --------                          ----------
# protocol  num aliases      # comments
hopopt        0 HOPOPT       # IPv6 Hop-by-Hop Option          [RFC1883]
icmp          1 ICMP         # Internet Control Message         [RFC792]
igmp          2 IGMP         # Internet Group Management       [RFC1112]
ggp           3 GGP          # Gateway-to-Gateway               [RFC823]
ip            4 IP           # IP in IP (encapsulation)        [RFC2003]
st            5 ST           # Stream                  [RFC1190,RFC1819]
tcp           6 TCP          # Transmission Control             [RFC793]
cbt           7 CBT          # CBT                            [Ballardie]
egp           8 EGP          # Exterior Gateway Protocol  [RFC888,DLM1]
igp           9 IGP          # any private interior gateway       [IANA]
bbn-rcc-mon  10 BBN-RCC-MON  # BBN RCC Monitoring                  [SGC]
nvp-ii       11 NVP-II       # Network Voice Protocol       [RFC741,SC3]
pup          12 PUP          # PUP                           [PUP,XEROX]
argus        13 ARGUS        # ARGUS                               [RWS4]
emcon        14 EMCON        # EMCON                                [BN7]
xnet         15 XNET         # Cross Net Debugger          [IEN158,JFH2]
chaos        16 CHAOS        # Chaos                                [NC3]
udp          17 UDP          # User Datagram                [RFC768,JBP]
```

. . .

Servicios de red xinetd

1. Presentación

El demonio **xinetd** es un "superservicio" que permite controlar el acceso a un conjunto de servicios,**telnet** por ejemplo. Se pueden configurar muchos servicios de red para funcionar con **xinetd**, como los servicios ftp, ssh, samba, rcp, http, etc. Se pueden aplicar opciones de configuración específicas para cada servicio gestionado.

Cuando un cliente se conecta a un servicio de red controlado por **xinetd**, xinetd recibe la petición y verifica primero las autorizaciones de acceso TCP (vea **tcp_wrappers** en el próximo capítulo); luego, las reglas definidas para este servicio (autorizaciones específicas, recursos asignados, etc.). El demonio levanta una instancia del servicio y le cede la conexión. A partir de entonces, **xinetd** ya no interfiere en la conexión entre el cliente y el servidor.

2. Configuración

Los archivos de configuración son:

- **/etc/xinetd.conf**: configuración global
- **/etc/xinetd.d/***: directorio que contiene los archivos específicos para los servicios. Existe un archivo por servicio, con el mismo nombre que el especificado en **`/etc/services`**.

```
$ ls -l /etc/xinetd.d
total 92
-rw-r--r-- 1 root root  313 sep 22  2007 chargen
-rw-r--r-- 1 root root  333 sep 22  2007 chargen-udp
-rw-r--r-- 1 root root  256 mar 20 22:11 cups-lpd
-rw-r--r-- 1 root root  409 nov  4  2005 cvs
-rw-r--r-- 1 root root  313 sep 22  2007 daytime
-rw-r--r-- 1 root root  333 sep 22  2007 daytime-udp
-rw-r--r-- 1 root root  313 sep 22  2007 discard
-rw-r--r-- 1 root root  332 sep 22  2007 discard-udp
-rw-r--r-- 1 root root  305 sep 22  2007 echo
-rw-r--r-- 1 root root  324 sep 22  2007 echo-udp
-rw-r--r-- 1 root root  492 sep 22  2007 netstat
-rw-r--r-- 1 root root  207 abr 23 19:04 rsync
-rw-r--r-- 1 root root  337 feb 17 14:22 sane-port
-rw-r--r-- 1 root root  332 sep 22  2007 servers
-rw-r--r-- 1 root root  334 sep 22  2007 services
-rw-r--r-- 1 root root  351 jun 21  2007 svnserve
-rw-r--r-- 1 root root  277 nov  8  2007 swat
-rw-r--r-- 1 root root  536 sep 21  2007 systat
-rw-r--r-- 1 root root  387 feb  4 10:11 tftp.rpmsave
-rw-r--r-- 1 root root  339 sep 22  2007 time
-rw-r--r-- 1 root root  333 sep 22  2007 time-udp
-rw-r--r-- 1 root root 2304 abr  4 11:39 vnc
-rw------- 1 root root  768 sep 22  2007 vsftpd
```

Contenido de xinetd.conf:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
```

```
}
includedir /etc/xinetd.d
```

instances: número máximo de peticiones que un servicio **xinetd** puede gestionar en un instante dado.

- **log_type**: en nuestro caso, el demonio **syslog** gestiona las trazas mediante **authpriv** y las trazas están colocadas en /var/log/secure. **FILE /var/log/xinetd** hubiera colocado las trazas en /var/log/xinetd.
- **log_on_success**: **xinetd** va a registrar el evento si la conexión al servicio tiene éxito. La información trazada son el cliente (**HOST**) y el **ID** del proceso servidor que trata de la conexión.
- **log_on_failure**: igual para los fracasos. Resulta fácil saber qué clientes han intentado conectarse si, por ejemplo, no se autoriza la conexión.
- **cps**: **xinetd** sólo autoriza 25 conexiones por segundo a un servicio. Si se alcanza el límite,**xinetd** esperará 30 segundos antes de autorizar de nuevo las conexiones.
- **includedir**: incluye las opciones de los archivos presentes en el directorio indicado.

Ejemplo /etc/xinetd.d/telnet:

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        disable         = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
}
```

La primera línea de comentario, **default**, tiene una importancia particular. No la interpreta **xinetd**, sino **ntsysv** o **chkconfig**, para determinar si el servicio está activo.

- **service**: nombre del servicio que corresponde a un servicio definido en **`/etc/services`**.
- **flags**: atributos para la conexión. REUSE indica que se volverá a utilizar el socket para una conexión telnet.
- **socket_type**: especifica el tipo de socket. En general, **stream** (tcp) o **dgram** (udp). Una conexión directa IP se hace por **raw**.
- **wait**: indica si el servidor es single-threaded (yes) o multi-threaded (no).
- **user**: con qué cuenta de usuario se iniciará el servicio.
- **server**: ruta del ejecutable que se debe iniciar.
- **log_on_failure**: el **+=** indica que se añade la opción asociada al archivo de traza, además de las opciones por defecto. Aquí: el login.
- **disable**: indica si el servicio está activo o no.

Algunas opciones pueden mejorar las condiciones de acceso y la seguridad:

- **only_from**: permite el acceso únicamente a los anfitriones especificados.
- **no_access**: impide el acceso a los anfitriones especificados (p. ej.: 172.16.17.0/24).
- **access_times**: autoriza el acceso únicamente en una franja horaria dada (p.ej.: 09:00-18:30).

3. Inicio y parada de los servicios

Se distinguen dos casos.

Primer caso, el servicio **xinetd** es un servicio como otro cualquiera cuyo inicio o cuya parada puede efectuarse con el comando **service** o directamente mediante la ejecución de /etc/init.d/xinetd.

```
# service xinetd start
```

En este caso, el comando **chkconfig** (Red Hat, openSUSE) autoriza o no el arranque del servicio al inicio para cada nivel de ejecución (runlevel).

```
# chkconfig --level 345 xinetd on
```

Segundo caso, como **xinetd** gestiona varios servicios, la parada de **xinetd** detiene todos los servicios asociados y el lanzamiento de **xinetd** inicia todos los servicios asociados. No es posible elegir qué servicios de xinetd se han iniciado en tal o cual nivel de ejecución. Pero puede elegir entre activar o desactivar simplemente un servicio con **chkconfig**.

```
# chkconfig telnet on
```

Conexión PPP

1. Elección y configuración del módem

a. El caso de los Winmodems

Linux soporta completamente todos los módems RTC (analógicos) que se conectan en un puerto serie (externo) o que emulan un verdadero puerto serie (tarjeta PCI o mediante el puerto USB).

Sin embargo, existe una categoría particular de módems llamados los **winmodems**. Se presentan a veces como módems "verdaderos" (en ocasiones se parecen). En general, huya de este tipo de módems. No obstante, algunos modelos son compatibles con Linux. Para obtener información sobre este tema, diríjase al sitio http://linmodems.org/.

Linux reconoce como módems otros adaptadores distintos de los módems RTC. Es el caso de algunos adaptadores ADSL, pero también de los teléfonos móviles conectados por un cable USB o mediante una conexión Bluetooth.

b. Los archivos periféricos

Los puertos serie de tipo RS232 se llaman **ttySn**:

- **/dev/ttyS0**: primer puerto serie.
- **/dev/ttyS1**: segundo puerto serie.
- etc.

Los puertos serie de tipo USB se llaman **ttyUSBn: /dev/ttyUSB0**, y así sucesivamente.

Los puertos serie de comunicación mediante bluetooth se llaman **rfcommn** (para radio frequency communication): **/dev/rfcomm0**, y así sucesivamente.

Para utilizar los puertos serie y establecer una comunicación, debe poder escribir en los periféricos (para mandar órdenes) y, así, o bien conocer los derechos correspondientes, o bien utilizar un programa SUID, o aun disponer de reglas udev adaptadas.

c. Ajustar el puerto serie

Los puertos serie se gestionan mediante el comando **setserial**.

El comando **setserial** permite consultar la configuración de un puerto serie con el parámetro `-g`. El puerto serie ttyS0 es de tipo 16550A (el más rápido), utiliza el IRQ 4 y el puerto de dirección 0x03f8.

```
# setserial -g /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Como la información no es suficiente, añada el parámetro `-a`. Obtiene entre otras cosas la velocidad de la línea, que aquí es de 115200 bits por segundo (se debe proscribir la palabra "baudio").

```
# setserial -a -g /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
        Baud_base: 115200, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test
```

Setserial permite también configurar el puerto. Al consultar el puerto, puede saber qué parámetros

han permitido su ajuste con –G:

```
# setserial -G /dev/ttyS0
/dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_normal
skip_test
```

De ahí es posible extrapolar nuevos valores. Por ejemplo, pase el puerto serie a una velocidad de 57600 bps:

```
# setserial /dev/ttyS0 baud_base 57600
# setserial -a -g /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
        Baud_base: 57600, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test
```

d. Los comandos AT

Todos los módems utilizan un par de comandos estándares llamados **comandos AT**. Su verdadero nombre es **Comandos Hayes**, del nombre de la empresa que los inventó. AT significa Attention (Cuidado). Después de estas primeras letras, el módem espera una serie que permita configurarlo, numerarlo, colgarlo, etc.

El par suele ser estándar, pero la configuración cambia de un modelo a otro. Si no quiere entrar en detalles, una configuración genérica es suficiente y funciona para la casi totalidad de los módems. Como no es posible describir los comandos AT, encontrará una lista en el sitio 3com (que ha comprado US Robotics, el mejor fabricante de módems):http://www.usr.com/support/3cp3056/3cp3056-spanish-ug/atcoms.htm

Sin embargo, veamos algunos:

- Marcar: ATDT0102030405
- Contestar: ATA
- Colgar: ATH

2. PPP

El protocolo **PPP** (*Point to Point Protocol*) le permite relacionarse con otra máquina con el fin de acceder a ella y a su red, lo que suele ser el caso de Internet. Hoy en día, y a pesar de las numerosas soluciones propuestas por el cable o el ADSL, algunas conexiones se siguen haciendo por módem RTC (módem clásico conectado en un puerto USB, serie o interno) conectado a un enchufe de los de siempre.

Para establecer una conexión PPP necesita:

- Un cliente que disponga de las herramientas ppp (pppd) y chat para dialogar con el servidor.
- Un servidor que disponga de pppd y medios para facilitar una dirección IP (dhcp).
- Un módem.

Lo que viene a continuación solo tiene en cuenta la parte cliente.

Debe conocer:

- El puerto serie en el cual está conectado su módem: ttySX (serie o USB), ttyACMX (USB), rfcommX (Bluetooth), etc.
- El número de teléfono de su proveedor de acceso a Internet (PAI).
- El número de usuario y la contraseña en su PAI.

- La dirección del servidor DNS de su PAI.

El módem no tiene por qué ser RTC. Un móvil reconocido como módem mediante el cable de conexión al PC o con el protocolo Bluetooth sirve perfectamente de módem. Para que sea conforme a las normas 4G, 3G o Edge, los flujos pueden ser muy impresionantes. Un gran número de conexiones ADSL que atraviesan los módems se efectúan también mediante el protocolo PPP. En este caso, la serie se aplica, pero se deben prever unas modificaciones.

3. Conexión mediante la consola

a. Manualmente

En el ejemplo siguiente:

- El número de teléfono es 0102030405.
- El login es "login".
- La contraseña es "password".
- El periférico es /dev/modem.

Aunque DHCP pueda gestionarlo en el momento de la conexión, puede ser necesario modificar el archivo /etc/resolv.conf para indicar los servidores de nombres (DNS) de su proveedor.

La conexión PPP necesita que el servicio **pppd** (en general, `/usr/sbin/pppd`) sea ejecutado como root. Para ello, se suele ubicar el derecho SUID:

```
-rwsr-xr-t 1 root root 316392 2008-04-04 19:03 pppd*
```

Otra solución es dar estos derechos a la herramienta de conexión (chat, kppp, etc.) o modificar en consecuencia los derechos de los periféricos (es el caso de varias distribuciones).

Los archivos de configuración se ubican en `/etc/ppp`:

```
# ls -l /etc/ppp
total 48
-rw------- 1 root root  690 sep 21  2007 chap-secrets
-rw-r--r-- 1 root root  449 sep 21  2007 filters
lrwxrwxrwx 1 root root    5 may  9 20:47 ip-down -> ip-up
drwxr-xr-x 2 root root 4096 sep 21  2007 ip-down.d
-rwxr-xr-x 1 root root 6175 abr 24 00:26 ip-up
drwxr-xr-x 2 root root 4096 sep 21  2007 ip-up.d
-rw-r--r-- 1 root root 7943 sep 21  2007 options
-rw------- 1 root root  340 sep 21  2007 options.pptp
-rw------- 1 root root 1219 sep 21  2007 pap-secrets
drwxr-xr-x 2 root root 4096 feb 23 23:18 peers
-rwxr-xr-x 1 root root 3778 abr 24 00:26 poll.tcpip
```

Veamos un ejemplo de conexión a un servidor PPP:

```
#/bin/sh
/usr/sbin/pppd connect '/usr/sbin/chat -v ABORT ERROR ABORT "NO CARRIER" \
    ABORT BUSY  "" ATZ OK ATDT0102030405 CONNECT "" ogin: "login" \
    word: "password"' \
    /dev/modem 38400 noipdefault debug crtscts modem defaultroute &
```

b. Por los archivos

Va a necesitar dos archivos. El primero contendrá los comandos del servicio pppd; el segundo, la secuencia de comunicación con el PAI. Los dos están en /etc/ppp/peers.

Veamos el primer archivo /etc/ppp/peers/cnx1:

```
# cat /etc/ppp/peers/cnx1
/dev/modem
connect '/usr/sbin/chat -v -f /etc/ppp/peers/cnx1-chat'
defaultroute
noipdefault
usepeerdns
115200

debug
noauth

maxfail 10
lcp-echo-interval 5
lcp-echo-failure 12
holdoff 3
noaccomp noccp nobsdcomp nodeflate nopcomp novj novjccomp
lock
crtscts
```

Cada línea contiene al menos una instrucción; presentamos las más pertinentes:

- **/dev/modem**: el periférico de conexión (el módem);
- **connect**: la cadena de conexión mandada al PAI;
- **defaultroute**: la ruta facilitada por el PAI sustituye a la ruta por defecto;
- **noipdefault**: el PAI provee el IP por su DHCP;
- **usepeerdns**: recupera la información DNS del PAI;
- **115200**: la velocidad de comunicación del periférico (se negociará);
- **debug**: provee el detalle completo de la conexión;
- **noauth**: no es el script ppp que establece la autenticación (ver línea connect);
- **maxfail**: n intentos de conexión antes de abandonar;
- **holdoff**: espera de n segundos entre dos conexiones;
- **lock**: permite el acceso exclusivo al archivo periférico;
- **crtscts**: activa el control de flujo material.

Veamos el segundo archivo, /etc/ppp/peers/cnx1-chat, utilizado por la línea **connex** del primer archivo, /etc/ppp/peers/cnx1:

```
# cat /etc/ppp/peers/cnx1-chat
ABORT ERROR
ABORT "NO CARRIER"
ABORT BUSY  "" ATZ
OK ATDT0102030405
CONNECT ""
ogin: "login"
word: "password"
```

Sólo se trata de un ejemplo. Tiene que comprobar, tanto por parte de su PAI como en la documentación de su módem, cuáles son los comandos correctos AT que hay que pasar (suelen ser estándar).

c. Conexión

Inicialice la conexión:

```
# pppd call cnx1
```

Debería ver las leds de su módem parpadear y, si el altavoz está activado, se oye el pitido característico. Si se establece la conexión, una nueva interfaz de red aparece: **ppp0**.

```
# ifconfig ppp0
ppp0      Link encap:Point-Point Protocol
          inet addr:10.xx.yy.zz  P-t-P:10.xx.yy.zz Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Las distribuciones generalmente vienen acompañadas de herramientas de tipo Network Manager que permiten establecer una conexión ethernet, Wi-Fi o ppp en un instante con algunos clics. La distribución Mandriva merece una mención especial: la configuración de una conexión Internet mediante un móvil 3G a través del protocolo Bluetooth se hace en unos segundos, con la herramienta Drakconf que lo detecta todo por sí misma.

OpenSSH

1. Presentación

OpenSSH es un protocolo de shell con prestaciones de seguridad, un mecanismo que permite la autenticación segura, la ejecución remota y la conexión a distancia. Permite también el transporte seguro del protocolo X Window. En realidad, es capaz encapsular protocolos no seguros redireccionando los puertos.

Los paquetes que se deben utilizar para un servidor son **openssh**, **openssl** y **openssh-clients**. Para X se añaden los paquetes **openssh-askpass*** (puede haber varios en función del entorno de escritorio). La lista de los paquetes que hay que instalar depende de cada distribución.

El uso más común sigue siendo el acceso remoto seguro a una máquina mediante el cliente ssh.

2. Configuración

La configuración es /etc/ssh/sshd_config. Si es preciso, se pueden modificar algunas opciones:

- **Port**: el número de puerto por defecto es 22;
- **Protocol**: fijado en 2,1, autoriza SSH1 y SSH2. Se preferirá SSH2 y así se dejará el valor 2;
- **ListenAddress**: por defecto ssh escucha en todos los IP del servidor. Se puede autorizar únicamente la escucha en una interfaz dada;
- **PermitRootLogin: ssh** autoriza las conexiones de root. Se puede poner el valor a "**no**". En este caso, habrá que conectarse como simple usuario y pasar por **su** o **sudo**;
- **Banner**: ruta de un archivo cuyo contenido se mostrará a los usuarios durante la conexión.

ssh es un servicio System V que puede iniciarse con service o directamente por /etc/init.d/sshd.

```
# service sshd start
```

3. Utilización

El comando **ssh** permite establecer una conexión.

```
$ ssh -l login host
$ ssh login@host
```

La opción –X permite activar la redirección (forwarding) del protocolo X Window.

```
$ ssh -X login@host
```

4. Claves y conexión automática

Es posible establecer una conexión automática hacia otra máquina sin introducir una contraseña. Para ello, es necesario generar un par de claves, privada y pública, desde la cuenta de usuario del cliente (la máquina que se va a conectar). No se debe introducir ninguna frase de contraseña.

Por parte del servidor ssh, se debe colocar la clave pública del cliente en un archivo que contiene las claves autorizadas para conectarse en la cuenta de destino.

a. Lado cliente

- Genere una clave en formato RSA con el comando **ssh-keygen**:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bean/.ssh/id_rsa):
Created directory '/home/bean/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bean/.ssh/id_rsa.
Your public key has been saved in /home/bean/.ssh/id_rsa.pub.
The key fingerprint is:
f6:39:23:4e:fa:53:d0:4e:65:7f:3f:fd:a3:f4:8e:2a bean@p64p17bicb3
```

- El directorio del usuario contiene ahora un directorio .ssh:

```
$ cd .ssh
$ ls
id_rsa id_rsa.pub
```

- El archivo id_rsa.pub contiene la clave pública:

```
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArB/VskR9v708J2EDG1LM1Q6HmKJc
P2UenurnSr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQE
hhPLwJVLXbGNc8lMf742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2q
zKKx7ZuNQEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzfk
vu7zQbOGDGGPiF3aOvd0oSBNgiJtZ+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3T
Q4xU6WxkxqTBcsjEm1gIymFAyxDo+zzf63jxLGO8Pp50DKf7DUqBx7+rjw==
bean@slyserver
```

b. Lado servidor

- Vaya al directorio .ssh de la cuenta a la cual desea acceder en el servidor (cree una si no existe):

```
$ cd /home/seb/.ssh
```

- Edite el archivo authorized_keys2 (creélo si no existe) y copie dentro en una nueva línea el contenido del archivo id_rsa.pub del cliente. Guarde.

```
$ echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEArB/VskR9v708J2EDG1LM1Q6HmKJcP2Uenurn
Sr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQEhhPLwJVL
XbGNc8lMf742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2qzKKx7ZuN
QEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzfkvu7zQbOG
DGGPiF3aOvd0oSBNgiJtZ+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3TQ4xU6Wxk
xqTBcsjEm1gIymFAyxDo+zzf63jxLGO8Pp50DKf7DUqBx7+rjw== bean@slyser
ver" >> autorized_keys2
```

- Intente una conexión; no se requiere la contraseña:

```
$ ssh seb@slyserver
```

c. Copia automática

Si bien el método anterior es didáctico, no es nada práctico. Utilice siempre el comando **ssh-copy-id**:

```
$ ssh-copy-id ~/.ssh/id_rsa.pub seb@slyserver
```

Se solicita la contraseña de la cuenta seb (la primera vez) luego la clave se copia en el archivo de claves autorizadas del host remoto.

5. Passphrase y agente SSH

Además de la seguridad de la clave SSH y el cifrado del flujo de datos, debe considerar el uso de una frase de paso, que se solicita durante la conexión SSH. Si su clave pública no es reconocida por el host, se solicitará la frase y contraseña. Si olvidó a introducir una contraseña al crear la clave, o desea cambiarla, utilice la opción `-p` de ssh-keygen:

```
$ ssh-keygen -p
Enter file in which the key is (/home/seb/.ssh/id_rsa):
Key has comment '/home/seb/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

La introducción de su frase con cada conexión SSH puede convertirse rápidamente en un fastidio. Muchos SO o entornos de escritorio ofrecen almacenarla en sus herramientas de seguridad mientras dure la sesión o de forma permanente en coordinación con el comando **ssh-agent**.

ssh-agent es un servicio que almacenará sus claves privadas y las frases correspondientes durante la sesión, o un tiempo delimitado a escoger. Generalmente lo arranca la sesión gráfica, pero también se puede iniciar de forma manual:

```
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-q0vj5zBtrC1Y/agent.2118; export SSH_AUTH_SOCK;
SSH_AGENT_PID=2119; export SSH_AGENT_PID;
echo Agent pid 2119;
```

Seleccione las últimas líneas. Haga un copiar y pegar y ejecútelas (sobre todo la parte SSH_AUTH_SOCK).

```
$ SSH_AUTH_SOCK=/tmp/ssh-q0vj5zBtrC1Y/agent.2118; export SSH_AUTH_SOCK;
```

Ahora puede iniciar el comando ssh-add para añadir su passphrase:

```
$ ssh-add
Enter passphrase for /home/seb/.ssh/id_rsa:
Identity added: /home/seb/.ssh/id_rsa (/home/seb/.ssh/id_rsa)
```

La passphrase no volverá a ser solicitada durante la duración de la sesión.

Si desea eliminar una identidad antes del fin de la sesión, por ejemplo si se bloquea antes de desconectar, emplee `-d`:

```
$ ssh-add -d
Identity removed: /Users/seb/.ssh/id_rsa ( seb@linux-h7e0.site)
```

Montar un servidor DHCP

1. Presentación

El servicio **DHCP** (*Dynamic Host Configuration Protocol*), protocolo de configuración dinámico de los anfitriones, permite a los anfitriones de una red pedir y recibir información de configuración (dirección, encaminamiento, DNS, etc.). Suele haber un solo servidor DHCP por segmento de red incluso si hay varios posibles. Si el servidor está en otro segmento, se puede utilizar un agente de retransmisión DHCP.

Dicho de otro modo, un cliente DHCP busca sólo un servidor DHCP, que le comunicará su dirección IP. Se asigna la dirección IP de manera dinámica a partir de rangos de direcciones predefinidas, o de manera estática en función de la dirección MAC del peticionario. La información es válida durante un período de tiempo dado (un contrato) que se puede renovar y configurar.

DHCP es un programa basado en **BOOTP** (*Bootstrap Protocol*). Cuando el cliente busca contactar un servidor, BOOTP le facilita la información de direccionamiento. DHCP gestiona las renovaciones. BOOTP se basa en el protocolo de transporte UDP.

Un cliente del servicio no tiene información de red disponible en el momento del inicio. Debe encontrar sólo un servidor DHCP. Para ello, BOOTP efectúa un broadcast mediante la IP 255.255.255.255 con una trama que contiene sus datos (como su dirección MAC) y los datos deseados (tipo de petición, aquí DHCPDISCOVER, puerto de conexión, etc.). Se manda el broadcast por definición a todos los anfitriones de la red local. Cuando el servidor DHCP detecta la trama, efectúa también un broadcast (el anfitrión cliente aún no tiene IP) con la información básica que desea el anfitrión (DHCPOFFER, primeros parámetros). El anfitrión establece una primera configuración; luego pide confirmación de la IP (DHCPREQUEST).

El servidor DHCP confirma (DHCPHACK). El contrato está confirmado y el cliente dispone entonces de toda la información válida.

2. Arranque del servidor dhcpd

El servidor **dhcpd** es un servicio (daemon) iniciado mediante un script (/etc/init.d/dhcpd). Se configura con el archivo /etc/dhcpd.conf. Las direcciones IP asignadas se colocan en /var/lib/dhcp/dhcpd.leases.

```
# service dhcpd start
```

o:

```
# /etc/init.d/dhcpd start
```

3. Información básica

El archivo de configuración de un servidor es bastante simple si se limita a unos ajustes básicos.

```
ddns-update-style none; # ninguna actualización del DNS par DHCP
option domain-name "pepito.es"; # nombre de dominio transmitido al cliente
option domain-name-servers 192.168.1.254; # lista de los DNS separados
con comas
default-lease-time 21600; # duración del arrendamiento por defecto en segundos
sin petición explícita
max-lease-time 43200; # duración máx del arrendamiento si la petición del cliente
es más elevada
```

Como dhcpd puede gestionar varias subredes, se debe especificar las reglas que hay que aplicar para cada subred. En general, en el contexto de una pequeña red, sólo habrá un único bloque, pero

se pueden considerar todos los casos. Si está seguro de que sólo tiene una red, puede omitir la declaración de la subnet (subred).

```
# Gestión de la subred 192.168.1.0
subnet 192.168.1.0 netmask 255.255.255.0
{
  option routers 192.168.1.254; # pasarela para esta red
  option subnet-mask 255.255.255.0; # máscara de subred
  range 192.168.1.2 192.168.1.250; # Configuración del rango DHCP

  # Caso de asignación de IP estáticas
  fuera puesto1
  {
    hardware ethernet 00:A0:ad:41:5c:b1; # Dirección MAC
    fixed-address 192.168.1.1; # esta máquina tendrá la IP 192.168.1.1
  }
```

Algunos clientes DHCP ignoran totalmente el hecho de que un servidor DHCP puede asignar un nombre (hostname) al anfitrión de manera dinámica. En el ejemplo anterior, la máquina con el IP 192.168.1.1 debería obtener el nombre puesto1. Ahí tiene un ejemplo:

```
  # los clientes recibirán los nombres de los host declarados
  use-host-decl-names on;
  host puesto1
  {
    hardware ethernet 00:A0:ad:41:5c:b1; # Dirección MAC
    fixed-address 192.168.1.1; # esta máquina tendrá la IP 192.168.1.1
y el nombre puesto1
  }
```

También puede trabajar caso por caso:

```
  host puesto2
  {
    hardware ethernet 00:A0:ad:41:5c:b2; # Dirección MAC
    fixed-address 192.168.1.251; # este host tendrá la IP 192.168.1.251
    option host-name "puesto2"; # este host tendrá como nombre puesto2
  }
```

4. Cliente

En Linux y las distribuciones de tipo Red Hat, Fedora, Mandriva, openSUSE, etc., modifique el archivo **/etc/sysconfig/network-script/ifcfg-xxx** especificando dhcp para **BOOTPROTO** y vuelva a iniciar la conexión de red (**ifdown**, luego **ifup**).

En Ubuntu y Debian, modifique la declaración de la interfaz de red en **/etc/network/interfaces**reemplazando static por dhcp:

```
Iface eth0 inet dhcp
```

El cliente dhcpcd permite activar dhcp en una interfaz de red. Por ejemplo, el método más sencillo para eth0 es:

```
# dhcpcd eth0 &
```

Puede transmitir opciones a dhcpcd para que se encargue de varias posibilidades. Entre estas opciones tenemos:

- -D: autoriza la modificación del nombre de dominio;

- `-H`: autoriza la modificación del nombre de anfitrión;
- `-R`: evita la sobreescritura del archivo resolv.conf;
- `-l`: permite modificar el período de vigencia (leasetime) en segundos.

```
# dhcpcd -D -H -l 86400 eth0
```

Servidor DNS

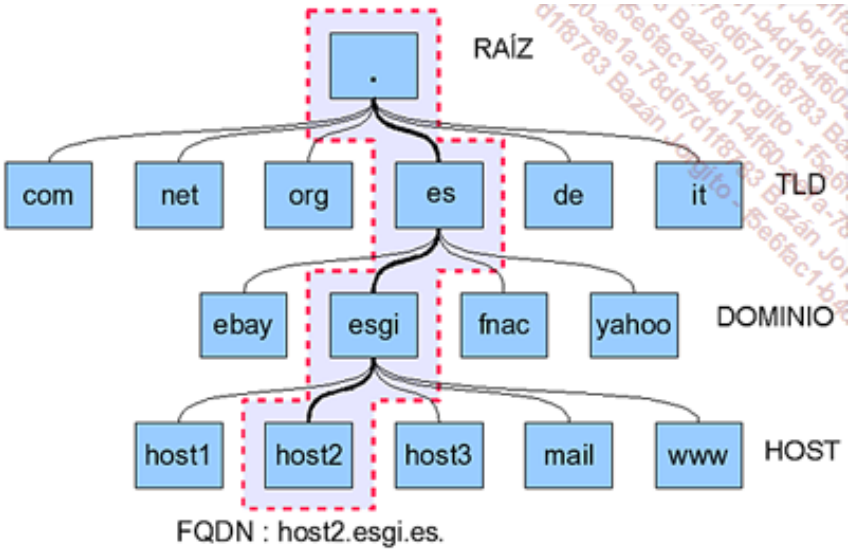
1. Presentación

El sistema de Nombres de Dominios **DNS** (*Domain Name System*) transforma los nombres de anfitrión en direcciones IP: es la **resolución de nombre** y transforma las direcciones IP en nombres de anfitrión: es la **resolución inversa**. Permite agrupar las máquinas por dominios de nombre. Facilita información de encaminamiento y correo electrónico.

El DNS permite referirse a sistemas basados en IP (los anfitriones) empleando nombres amenos (los*nombres de dominios*). El interés de un DNS es obvio. Los nombres de dominio son más fáciles de recordar y, si su dirección IP cambia, el usuario no se da cuenta. Por lo tanto, se entiende que el DNS es un servicio fundamental para Internet.

Los nombres de dominio van separados por puntos. Como cada elemento se puede componer de 63 caracteres, sólo puede haber un máximo de 127 elementos y el nombre completo no debe superar los 255 caracteres. El nombre completo no abreviado se llama **FQDN** (*Fully Qualified Domain Name*). En un FQDN, el elemento más a la derecha se llama **TLD** (*Top Level Domain*), el más a la izquierda representa al anfitrión y, por tanto, la dirección IP.

El DNS contiene una configuración especial para los enrutadores de correo electrónico (definiciones MX) que permiten una resolución inversa, un factor de prioridad y tolerancia a errores.



Representación de un árbol DNS

Una zona es una parte de un dominio gestionado por un servidor particular. Una zona puede gestionar uno o varios subdominios y se puede repartir un subdominio en varias zonas. Una zona representa una unidad de administración, de la cual puede ser responsable una persona.

2. Inicio

El servicio se llama **named**.

```
# service named start
```

o:

```
# /etc/init.d/named start
```

3. Configuración de Bind

Bind (*Berkeley Internet Name Daemon*) es el servidor de nombres más utilizado en Internet. Bind 9 soporta el IPv6, los nombres de dominio unicode, el multithread y muchas mejoras de seguridad.

a. Configuración general

La configuración global de Bind se encuentra en el archivo **`/etc/named.conf`**. La configuración detallada de las zonas se coloca en `/var/lib/named`. `/etc/named.conf` se compone de dos partes. La primera corresponde a la configuración global de las opciones de Bind. La segunda es la declaración de las zonas para los dominios individuales. Los comentarios empiezan con una # o //.

Cuidado: a veces ocurre (en particular en RHEL 4.x) que la configuración de Bind sea "chrooteada" (movida a una estructura específica de la cual no puede salir el servicio, que tampoco puede acceder al resto de la estructura). En CentOS y RHEL 4.x y superiores, named.conf está en `/var/named/chroot/etc/`. Se puede modificar este modo cambiando el archivo de configuración `/etc/sysconfig/named`.

```
# cat /etc/sysconfig/named
...
CHROOT=/var/named/chroot
...
```

En este caso, todos los archivos de configuración, incluidas las zonas, son relativos a esta ruta. Veamos un archivo named.conf básico.

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};
zone "localhost" in {
    type master;
    file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};
zone "." in {
    type hint;
    file "root.hint";
};
```

b. Sección global

La configuración global está ubicada en la sección **options**. Veamos un detalle de algunas opciones importantes (se debe especificar el punto y coma):

- **directory "filename";**: ubicación de los archivos que contienen los datos de las zonas.
- **forwarders { direccion-ip; };**: si el propio servidor bind no puede resolver la petición, se la manda a un servidor DNS exterior, por ejemplo el del proveedor de acceso.
- **listen-on-port 53 {127.0.0.1; direccion-ip; };**: puerto de escucha del DNS, seguido de las direcciones de escucha. Aquí se indican las direcciones IP de las interfaces de red de la máquina. No hay que olvidar 127.0.0.1.
- **allow-query { 127.0.0.1; red; };**: máquina(s) o red(es) autorizadas a utilizar el servicio DNS. Por ejemplo, 192.168.1/24. Si la directiva no está, se autoriza todo.

- **allow-transfer { 192.168.1.2; };**: máquina(s) o red(es) autorizadas a copiar la base de datos cuando hay una relación maestro y esclavo. Por defecto, no se autoriza ninguna copia.
- **notify no;**: se notifica o no a los demás servidores DNS un cambio en las zonas o un nuevo arranque del servidor.

c. Sección de zonas

Para cada dominio o subdominio, se definen dos secciones **zona**. La primera contiene los datos de resolución de nombre (Nombre hacia IP) y la segunda, los datos de resolución inversa (IP hacia Nombre). En cada caso, la zona puede ser maestro (**Master**) o esclavo (**Slave**):

- **Master**: el servidor contiene la totalidad de los registros de la zona en sus archivos de zona. Cuando recibe una petición, busca en sus archivos (o en su caché) la resolución.
- **Slave**: el servidor no contiene por defecto ningún registro. Se sincroniza con un servidor maestro del cual recupera toda la información de zona. Se puede colocar esta información en un archivo. En este caso, el esclavo almacena una copia local de la base de datos. Durante la sincronización, el número de serie de esta copia se compara con el del maestro. Si los números son diferentes, se hace una nueva copia; si no es el caso, se sigue utilizando la anterior.

d. Zona de resolución

Se suele llamar **zona**. Para cada dominio o subdominio, indica en qué archivo se coloca la información de la zona (o sea, entre otras cosas, las direcciones IP asociadas a cada anfitrión), su tipo (maestro o esclavo), si se autoriza o no la notificación, la dirección IP del servidor DNS maestro en el caso de un esclavo, etc.

El nombre de la zona es muy importante, ya que determina el dominio de búsqueda. Cuando el DNS recibe una petición, busca una correspondencia en todas las zonas.

```
zona "dominio.org" {
    type     "master";
    file     "dominio.org.zona";
};
```

- **type**: master o slave;
- **file**: nombre del archivo que contiene los datos de la zona. No hay reglas precisas para atribuir nombres, pero por razones de legibilidad se aconseja darle el mismo nombre que la zona tanto para una zona maestra como para una esclava. Para un master, es el original completado por usted si es preciso. Para un esclavo no es obligatorio. Si está presente, será una copia del master, sincronizada.
- En el caso de un Master, se puede añadir **allow-transfer** (servidores autorizados a duplicar la zona) y **notify yes** (indica una actualización o un nuevo arranque para los esclavos).

En caso de Slave: se añade la directiva **masters** para indicar a partir de qué servidor Master duplicar.

e. Zona de resolución inversa

Para cada red o subred IP (o rango de direcciones) se define una zona de resolución inversa cuyo archivo contiene una asociación IP hacia el nombre de máquina. Resulta ser casi lo mismo que la zona de resolución, excepto que se debe respetar una convención de atribución de nombres:

- El nombre de la zona termina siempre por un dominio especial **`.in-addr.arpa`**.
- Primero se debe determinar qué red debe cubrir la zona (caso de las subredes). Para

nosotros: una red de clase C 192.168.1.0, o sea **192.168.1/24**.

- Se invierte el orden de los bytes en la dirección: **1.168.192**.
- Se añade **in-addr.arpa**. Así, nuestro nombre de zona será **1.168.192.in-addr.arpa**.
- Para el resto se aplican las mismas observaciones que para la zona de resolución.

```
Zona "1.168.192.in-addr.arpa" {
    type    master;
    file    "192.168.1.zona";
};
```

f. Ejemplo

Supongamos un dominio (**dominio.org**) en una red de clase C 192.168.1.0. Supongamos dos servidores DNS 192.168.1.1 Master y 192.168.1.2 Slave.

En el Master

```
zona "dominio.org" {
    type    master;
    file    "dominio.org.zona";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
zona "1.168.192.in-addr.arpa" {
    type    master;
    file    "192.168.1.zona";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
```

En el Slave

```
zona "dominio.org" {
    type        slave;
    file        "dominio.org.zona";
    masters     { 192.168.1.1; };
};
zona "1.168.192.in-addr.arpa" {
    type        slave;
    file        "192.168.1.zona";
    masters     { 192.168.1.1; } ;
};
```

g. Zonas especiales

La zona raíz "." permite especificar los servidores raíces. Cuando ninguna zona consigue resolver una petición, se utiliza la zona raíz por defecto y ésta reenvía a los servidores raíces.

La zona de loopback no es necesaria, aunque sí útil. Sirve de **caché DNS**. Cuando una petición llega al servidor y éste no posee la información de resolución, la va a pedir a los servidores DNS raíces, que bajarán la información. Se coloca entonces esta información en caché. ¡Por lo tanto, los accesos siguientes serán mucho más rápidos!

4. Archivos de zonas

a. Definiciones

Los archivos de zonas utilizan varios términos, caracteres y abreviaciones específicas.

- **RR**: *Resource Record*. Nombre de un registro DNS (los datos del DNS).
- **SOA**: *Star Of Authority*. Permite describir la zona.
- **IN**: *the Internet*. Define una clase de registro que corresponde a los datos de Internet (IP). Si no se especifica para los registros, será la que haya por defecto.
- **A**: *Address*. Permite asociar una dirección IP a un nombre de anfitrión. Para IPv6 es AAAA.
- **NS**: *Name Server*. Designa un servidor DNS de la zona.
- **MX**: *Mail eXchanger*. Designa un servidor de correo electrónico con un indicador de prioridad. Cuanto más bajo es el valor, más elevada es la prioridad.
- **CNAME**: *Canonical Name*. Permite añadir alias: relacionar un nombre a otro. Se pueden crear alias sobre nombres de anfitrión y también sobre alias.
- **PTR**: *Pointer*. En una zona de resolución inversa, hace que una IP apunte a un nombre de anfitrión.
- **TTL**: *Time To Live*. Duración de vida de los registros de la zona.
- **@**: en las declaraciones de la zona es un alias (carácter de sustitución) para el nombre de la zona declarada en /etc/named.conf. Así, si la zona se llama dominio.org, @ vale dominio.org. En la declaración del administrador de la SOA, sustituye de manera puntual el punto en la dirección de correo electrónico.
- El punto "**.**": si se olvida el punto al final de la declaración de anfitrión, el nombre de la zona está concatenado al final del nombre. Por ejemplo, para la zona dominio.org, si se escribe**puesto1**, esto equivale a **puesto1.dominio.org**. Si se escribe **puesto1.dominio.org** (sin el punto final), entonces se obtiene el resultado siguiente: **puesto1.dominio.org.dominio.org**. Para evitarlo, debe escribir **puesto1.dominio.org**. (observe el punto al final).
- Algunos registros necesitan una noción de duración, que se suele expresar en segundos, pero también a veces con abreviaciones:
  - **1M**: un minuto, o sea, 60 segundos (1M, 10M, 30M, etc.);
  - **1H**: una hora, 3.600 segundos;
  - **1D**: un día, 86.400 segundos;
  - **1W**: una semana, 604.800 segundos;
  - **365D**: un año, 31.536.000 segundos.

Cuidado, esto es muy importante: en los archivos de zonas, **no se debe empezar una línea con espacios o tabuladores**. Esto no funciona: se interpretarían los espacios o tabulaciones como parte del nombre indicado, de la dirección o de la opción.

b. Zona

Empiece primero por una directiva **TTL**, que indica el tiempo de vida de la zona en segundos. Esto significa que cada registro de la zona será válido durante el tiempo indicado por **TTL** (nota: es posible modificar este valor para cada registro). Durante este tiempo, los otros servidores de nombres remotos pueden colocar los datos en caché. Un valor elevado permite reducir el número de peticiones efectuadas y alargar los plazos entre las sincronizaciones.

```
$TTL 86400
```

A continuación de las directivas TTL, registre el recurso **SOA**:

```
<domain>    IN    SOA    <primary-name-server> <hostmaster-email> (
                  <serial-number>
                  <time-to-refresh>
                  <time-to-retry>
                  <time-to-expire>
                  <minimum-TTL> )
```

- **domain**: es el nombre de zona, el mismo nombre utilizado en el archivo /etc/named.conf. • Está permitido sustituirlo por una @ si no se hace así, de lo contrario, no se debe olvidar finalizarlo con un punto (para evitar la concatenación).
- **primary-name-server**: el nombre del servidor primario en esta zona. No olvide declararlo en la lista de hosts (registros PTR o A).
- **hostmaster-email**: dirección de correo electrónico del administrador del servidor de nombres. Dado que la arroba @ ya está reservada para otro uso, se utiliza un punto para reemplazarlo. Así pues, "admin@dominio.org" debería escribirse "**admin.dominio.org.**".
- **serial-number**: es un número de serie que se debe incrementar manualmente en cada modificación del archivo de zonas con el objetivo de que el servidor de nombres sepa que debe volver a cargar esta zona. El número se utiliza en la sincronización con los servidores esclavos. Si el número de serie coincide con el de la última sincronización, los datos no necesitan ser actualizados. Por convención se usan diez cifras con el formato**YYYYMMDDNN** (año-mes-día-número).
- **time-to-refresh**: especifica a cualquier servidor esclavo cuánto tiempo debe esperar antes de consultar al servidor de nombres maestro si ha habido cambios en la zona.
- **time-to-retry**: indica al servidor esclavo cuánto tiempo debe esperar antes de emitir de nuevo una petición de actualización si el servidor maestro no ha respondido. La petición tendrá lugar cada time-to-retry segundos.
- **time-to-expire**: si, a pesar de las tentativas de contacto cada time-to-retry segundos, el servidor no ha respondido al final del período indicado por time-to-expire, el servidor esclavo cesa de responder a las peticiones para esta zona.
- **minimum-TTL**: el servidor de nombres solicita a los otros servidores de nombres poner en la caché la información para esta zona durante, al menos, el período indicado.

```
@    IN    SOA    dns1.dominio.org.    hostmaster.dominio.org. (
             2005122701    ; serial
             21600         ; refresh de 6 horas
             3600          ; intentar cada 1 horas
             604800        ; tentativas expiran tras una semana
             86400    )    ; TTL mínimo de un día
```

Diríjase a continuación a los registros **NS** (*Name Server*), donde especificará los servidores de nombres de esta zona.

```
    IN    NS    dns1
    IN    NS    dns2
```

Cuando no se especifique al comienzo de la línea un nombre de anfitrión o de zona (completo o con arroba @), querrá decir que se utilizará el mismo que el de la línea superior. En tanto en cuanto no se precise uno nuevo, será el último indicado el que se va a utilizar. Según lo afirmado anteriormente, las líneas podrían ser las siguientes:

```
@    IN    NS    dns1
@    IN    NS    dns2
```

o:

```
dominio.org. IN NS dns1
dominio.org. IN NS dns2
```

Observe la ausencia del punto tras el nombre de host; por ello, dominio.org está concatenado para obtener dns1.dominio.org.

```
IN NS dns1
```

equivale a:

```
IN NS dns1.dominio.org.
```

Diríjase a continuación a la relación de servidores de correo electrónico de zona. El valor numérico que aparece tras la MX establece la prioridad. Cuanto más bajo sea este valor, más prioritario será el servidor y más susceptible de ser contactado en primer lugar. Si los valores son idénticos, el correo es redistribuido de forma homogénea entre los servidores. Si un servidor no responde (sobrecargado, averiado) la petición bascula automáticamente a otra máquina.

```
    IN    MX    10    mail
    IN    MX    15    mail2
```

Si lo que desea es que una máquina responda pasando por el FQDN dominio.org sin precisar el anfitrión (por ejemplo, solicitar http://dominio.org en lugar de http://www.dominio.org), puede, en ese momento, declarar una dirección IP para dicho servidor. Así pues, el comando **ping dominio.org** responderá ¡192.168.1.3!

```
IN A 192.168.1.3
```

Puede declarar ahora otros anfitriones tales como servidores de nombres, de correo, de puestos, etc.

```
dns1        IN    A     192.168.1.1
dns2        IN    A     192.168.1.2
server1     IN    A     192.168.1.3
server2     IN    A     192.168.1.4
puesto1     IN    A     192.168.1.11
puesto2     IN    A     192.168.1.12
puesto3     IN    A     192.168.1.13
```

Vemos que no han sido declarados los servidores mail y mail2, y que tampoco se ha indicado el servidor web y ftp. Vamos a utilizar alias que apunten estos nombres de host a otros.

```
mail        IN    CNAME     server1
mail2       IN    CNAME     server2
www         IN    CNAME     server1
ftp         IN    CNAME     server1
```

Ha finalizado la configuración de la zona; ahora hay que ocuparse de la zona de resolución inversa.

c. Zona de resolución inversa

La zona de resolución inversa es casi idéntica a la anterior, sólo que los registros A son reemplazados por los registros PTR destinados a traducir una IP en un anfitrión. De ser posible, el TTL y la declaración SOA deben ser idénticos (excepto el nombre de zona). Además, debe poner los registros NS.

```
    IN    NS    dns1.dominio.org.
    IN    NS    dns2.dominio.org.
```

Usted no estará obligado a poner en la zona de resolución inversa la traducción de las direcciones IP del DNS, ¡ya que es el propio DNS el que resuelve su propio nombre! Sin embargo, hacerlo puede acelerar el proceso, ya que el DNS no debe ejecutar una consulta a sí mismo. Diríjase a los registros PTR traduciendo la dirección IP de cada anfitrión.

```
1     IN    PTR    dns1.dominio.org.
2     IN    PTR    dns2.dominio.org.
3     IN    PTR    server1.dominio.org.
4     IN    PTR    server2.dominio.org.
11    IN    PTR    puesto1.dominio.org.
12    IN    PTR    puesto2.dominio.org.
13    IN    PTR    puesto3.dominio.org.
```

Teóricamente, una misma IP puede atribuirse a varios anfitriones; los RFC no son muy explícitos sobre esta posibilidad, lo que, al final, puede crear problemas.

5. Diagnóstico de los problemas de configuración

El comando **named-checkconf** comprueba la sintaxis del archivo **named.conf**. Debe facilitarle el archivo como parámetro. La salida indicará las líneas que dan problemas.

El comando **named-checkzone** comprueba la sintaxis de un archivo de zona (incluso de resolución inversa). Debe especificarle el nombre del archivo de zona como parámetro.

6. Consulta dig y host

El programa **dig** es una herramienta de consulta avanzada de servidor de nombres capaz de restituir todos los datos de las zonas.

```
> dig tele2.es

; <<>> DiG 9.4.1-P1 <<>> tele2.es
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63972
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;tele2.es.                      IN      A

;; ANSWER SECTION:
tele2.es.               86363   IN      A       212.27.48.10

;; Query time: 1 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:36:09 2008
;; MSG SIZE  rcvd: 41
```

Por defecto, dig sólo restituye la dirección del host que se ha pasado como parámetro. En caso de éxito, el estado vale **NOERROR**; **ANSWER** indica el número de respuestas y dicha respuesta se sitúa debajo de la sección **ANSWER**. Para obtener una resolución inversa existen dos soluciones.

```
$ dig 10.48.27.212.in-addr.arpa ptr
```

o más sencillamente:

```
$ dig -x 212.27.48.10

; <<>> DiG 9.4.1-P1 <<>> -x 212.27.48.10
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60222
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;10.48.27.212.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
10.48.27.212.in-addr.arpa. 86400 IN     PTR     www.tele2.es.

;; Query time: 31 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:36:51 2008
;; MSG SIZE  rcvd: 68
```

En la primera sintaxis, observe que puede añadir un parámetro de consulta. Los principales se presentan a continuación.

- **a**: únicamente la dirección;
- **any**: toda la información correspondiente al dominio;
- **mx**: los servidores de mensajería;
- **ns**: los servidores de nombres;
- **soa**: la zona Start of Authority;
- **hinfo**: información sobre el anfitrión;
- **txt**: texto de descripción;
- **ptr**: zona reversa del anfitrión;
- **axfr**: lista de todos los anfitriones de la zona.

```
$ dig tele2.es any
; <<>> DiG 9.4.1-P1 <<>> tele2.es any
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28893
;; flags: qr aa; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 8

;; QUESTION SECTION:
;tele2.es.                      IN      ANY

;; ANSWER SECTION:
tele2.es.               86400   IN      NS      freens2-g20.tele2.es.
tele2.es.               86400   IN      A       212.27.48.10
tele2.es.               86400   IN      NS      freens1-g20.tele2.es.
tele2.es.               86400   IN      MX      20 mx2.tele2.es.
tele2.es.               86400   IN      SOA     freens1-g20.tele2.es.
hostmaster.proxad.net. 2008051001 10800 3600 604800 86400
tele2.es.               86400   IN      MX      10 mx1.tele2.es.

;; ADDITIONAL SECTION:
freens2-g20.tele2.es.   86400   IN      A       212.27.60.20
mx1.tele2.es.           86400   IN      A       212.27.48.6
mx2.tele2.es.           86400   IN      A       212.27.42.56
freens1-g20.tele2.es.   86400   IN      A       212.27.60.19
```

```
mx2.tele2.es.            86400   IN      A       212.27.42.58
mx1.tele2.es.            86400   IN      A       212.27.48.7
mx2.tele2.es.            86400   IN      A       212.27.42.57
mx1.tele2.es.            86400   IN      A       212.27.42.59

;; Query time: 9 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:35:32 2008
;; MSG SIZE  rcvd: 318
```

La herramienta **host** devuelve el mismo resultado, pero quizá de manera más sencilla.

```
$ host tele2.es
tele2.es has address 212.27.48.10
tele2.es mail is handled by 10 mx1.tele2.es.

$ host -t any tele2.es
tele2.es has address 212.27.48.10
tele2.es name server freens1-g20.tele2.es.
tele2.es has SOA record freens1-g20.tele2.es. hostmaster.proxad.net.
2008051001 10800 3600 604800 86400
tele2.es mail is handled by 10 mx1.tele2.es.

$ host -a tele2.es
Trying "tele2.es"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64513
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;tele2.es.                      IN      ANY

;; ANSWER SECTION:
tele2.es.               86140   IN      A       212.27.48.10
tele2.es.               86140   IN      NS      freens1-g20.tele2.es.
tele2.es.               86140   IN      SOA     freens1-g20.tele2.es.
hostmaster.proxad.net. 2008051001 10800 3600 604800 86400
tele2.es.               86140   IN      MX      10 mx1.tele2.es.

;; ADDITIONAL SECTION:
freens1-g20.tele2.es.   86140   IN      A       212.27.60.19
mx1.tele2.es.           86140   IN      A       212.27.48.7

Received 176 bytes from 10.23.254.240#53 in 4 ms
```

Correo electrónico

1. Fundamentos

- Cuando un cliente (un usuario) manda un mensaje, utiliza un **MUA** (*Mail User Agent*), por ejemplo Outlook Express, Thunderbird, Evolution, Kmail, Mutt, etc.
- El **MUA** envía el mensaje al **MTA** (*Mail Transport Agent*). El MTA estudia la dirección electrónica para aislar al usuario y dominio de destino. Luego verifica la información DNS de tipo **MX** (*Mail eXchanger*) para el dominio elegido, para saber a qué servidor transmitir el correo. Si ningún MTA está disponible, se coloca el mensaje en fila de espera y se reenvía más tarde (el plazo depende de la configuración del MTA).
- El MX puede ser o bien otro MTA, que tendrá el papel de enrutador (en caso de una redirección hacia un subdominio, por ejemplo) o bien un **MDA** (*Mail Delivery Agent*). El MDA coloca el mensaje en un archivo temporal, puede filtrarlo, etc.
- En este nivel, el destinatario recibe el mensaje: o lo recupera al leer directamente el archivo temporal (caso del comando mail, por ejemplo), o pasa por un protocolo de tipo **POP** o **IMAP**.
- El protocolo de transporte de mensajes es el **SMTP** (*Simple Mail Transfer Protocol*) en el puerto 25.
- Los protocolos de recepción de mensajes son o **POP** (*Post Office Protocol*) en el puerto 110 (POP3), o **IMAP** (*Internet Message Access Protocol*).

Dos suites de correo electrónico comparten la parte esencial del mercado en Unix: **sendmail** y**postfix**.

La serie libre **sendmail** es la más conocida y más utilizada. Eric Allman creó Sendmail en 1981, que fue integrado en BSD 4.2 en 1983. En 2000 se estimaba que lo usaban a más de 100 millones de servidores de correo electrónico. Mientras no haya que modificar su configuración básica, sendmail es idóneo. Si desea ir más lejos, le hará falta comprarse un libro completo sobre este tema. La configuración de sendmail es tan compleja que se inventó un lenguaje de macros llamado **m4** sólo para él. Así no se edita (o poco) el archivo de configuración de sendmail: se edita el archivo fuente de las macros y se vuelve a compilar: m4 va a crear el archivo de configuración de sendmail. Sendmail se ha convertido en un monstruo en lo que respecta a potencia y configuración.

El producto **postfix** tiende a ser cada vez más utilizado no tanto por aquellos a quienes sendmail ha decepcionado, sino por los que temen no saber configurarlo. Es una alternativa a sendmail. Los objetivos de sus desarrolladores (entre los cuales algunos son los de sendmail) son:

- la compatibilidad con sendmail;
- la rapidez (más de un millón de mensajes al día en un simple Pentium 4);
- la sencillez de administración (archivo de configuración simple y legible);
- la seguridad (se puede chrootear);
- la modularidad (descomposición de los tratamientos).

Nos quedaremos con la sencillez. En efecto, se puede configurar postfix con un solo comando sin necesitad de editar un archivo. Vamos a utilizar este servidor.

2. postfix

a. Configuración sencilla

La configuración de **postfix** se encuentra en /etc/postfix/main.cf. Se modifican sus valores o bien manualmente, o con la ayuda del comando **postconf**.

Postfix inicia primero un servicio maestro, **master**, encargado de los procesos

secundarios **smtpd**,**pickup** y **nqmgr**.

En algunas distribuciones, hay que modificar la configuración por defecto que utiliza la serie sendmail. Por ejemplo, en Red Hat debe indicar que se utilice postfix en lugar de sendmail con el comando **alternatives**.

```
#alternatives --set mta /usr/sbin/sendmail.postfix
```

Aplique una configuración básica con el comando **postconf**.

Dominio de origen de los mensajes

```
#postconf -e "myorigin = midominio.org"
```

De qué dominio recibir el correo

```
#postconf -e "mydestination = midominio.org"
```

De qué clientes transmitir el correo

```
#postconf -e "mynetworks = 192.168.1.0/24, 127.0.0.1"
```

En qué interfaces escuchar

```
#postconf -e "inet_interface = all"
```

Inicie el servicio.

```
#service postfix start
```

o:

```
#/etc/init.d/postfix start
```

b. Alias de usuarios

Es posible ubicar alias para los usuarios locales en el archivo **/etc/aliases**. Por ejemplo, si los mensajes de webmaster, admin y root se deben redireccionar hacia manuel:

```
manuel: webmaster, admin, root
```

c. Prueba

El registro se ubica en /var/log/maillog. Pruebe el servidor de la manera siguiente (por ejemplo):

```
mail -s `echo $USER` root@server1 < /etc/passwd
```

Si todo funciona, obtendrá trazas:

```
Fri 26 11:38:18 estacion1 postfix/pickup[12357] : F145040154: uid=0
from <root>
Fri 26 11:38:18 estacion1 postfix/cleanup[12318] : F145040154:
message-id=<20060126113017.F145040154:@estacion1.midominio.org>
Fri 26 11:38:26 estacion1 postfix/nqmgr[3469] : F145040154:
from=<root@estacion1.example.com>, size=314, nrcpt=1 (queue active)
Fri 26 11:38:32 estacion1 postfix/smtp[12468] : F145040154:
```

```
to=<root@server1>, relay=server1.midominio.org[192.168.1.1],
delay=17, status=sent (250 ok dirdel)
```

3. POP e IMAP

Existen varias suites para gestionar POP e IMAP. Una suite se llama **cyrus-imap** y en principio está reservada a las grandes estructuras y a los servidores 100 % dedicados al correo, o sea, donde los usuarios no se conectan.

Una solución se llama **dovecot**. Tras su instalación, basta con iniciarlo como servicio para que todo funcione, o casi todo.

Edite el archivo **`/etc/dovecot.conf`** para comprobar los protocolos soportados.

```
protocols = imap pop3
```

Inicie el servicio:

```
service dovecot start
```

o:

```
/etc/init.d/dovecot start
```

Pruebe a mandando un mensaje. Configure un cliente de mensajería para comprobar si el servidor POP funciona:

```
# telnet localhost 110
trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
+OK POP3 localhost.localdomain server ready
USER manuel
+OK user name accepted, password please
PASS password
+OK Mailbox open, 1 messages
STAT
+OK 1 384
TOP 1 99999
<mensaje aquí>
...
DELE 1
+OK Message deleted
QUIT
+OK bye
```

Servicio HTTP Apache

1. Presentación

Apache 2 es el servidor HTTP más utilizado actualmente en los servidores Web. Su configuración y su flexibilidad hacen que sea un servidor ineludible.

Cuando un servidor Apache recibe peticiones, puede distribuirlas a procesos hijos. La configuración permite iniciar procesos de manera anticipada y adaptar de manera dinámica este número según la carga.

Apache es modulable. Cada módulo permite añadir funcionalidades al servidor. El módulo más famoso es probablemente el que gestiona el lenguaje PHP, "mod_php". Cada módulo se añade mediante los archivos de configuración y no hace falta iniciar de nuevo el servidor Apache: sólo se le da la orden de volver a leer su configuración.

Apache puede gestionar varios sitios web a la vez, cada uno con su nombre, mediante los anfitriones virtuales.

2. Parada/Reinicio

El nombre del servicio depende de la distribución. A menudo se denomina **apache** o **httpd**. Según la distribución, inicie el servicio mediante el comando **service** o directamente por su nombre, /etc/init.d/apache.

- **/etc/init.d/httpd start**: inicia;
- **/etc/init.d/httpd stop**: detiene;
- **/etc/init.d/httpd restart**: reinicia;
- **/etc/init.d/httpd reload**: pide a Apache que vuelva a leer su configuración sin reiniciarse.

Apache va acompañado de la herramienta **apachectl**, que retoma los parámetros (lista no exaustiva) start, stop, status, reload, y sobre todo **configtest**, que valida o no el contenido del archivo de configuración de Apache.

3. Configuración

Se almacena la configuración principal en /etc/httpd/conf/httpd.conf: controla los parámetros generales del servidor web, los anfitriones virtuales y los accesos. La configuración de los diferentes módulos está ubicada en /etc/httpd/conf.d. Los módulos están presentes en /etc/httpd/modules/. Por defecto la raíz del servidor, donde se colocan las páginas del sitio, está en /var/www o /srv/www. Esta posición depende de la directiva **DocumentRoot** en los archivos de configuración.

4. Directivas generales

Es imposible listar todas las directivas del archivo httpd.conf, pero algunas son importantes.

- **ServerRoot**: directorio que contiene los archivos del servidor (configuración y módulos). Suele ser /etc/httpd.
- **Listen**: puertos en los cuales escucha el servidor Apache. Por defecto 80 (443 en https). Se pueden especificar varios puertos con varias directivas Listen. Si el servidor dispone de varias direcciones IP, se puede añadir la IP al puerto asociado: **Listen 192.168.1.3:80**.
- **User**: usuario de los procesos Apache. Nunca se utiliza root, sino una cuenta creada para la ocasión, en general Apache.

- **Group**: ídem, pero para el grupo.
- **ServerAdmin**: dirección de correo electrónico del administrador.
- **ServerName**: nombre de anfitrión (y puerto) del servidor. No corresponde obligatoriamente al nombre de anfitrión de la máquina. En cambio, es válido. **ServerName www.midominio.org**
- **UseCanonicalName**: si vale **on**, Apache va a responder utilizando la información de ServerName y Port, y no la información mandada por el cliente. Por ejemplo, un http//192.168.1.3 se transforma en http://www.midominio.org.
- **UserDir**: nombre de un subdirectorio donde cada usuario puede ubicar sus archivos HTML personales. En general, public_html. Se accede a él con:**http://www.midominio.org/~login/page.html**
- **ErrorLog**: archivo donde se colocan los logs de error del servidor.**`/var/log/httpd/error_log`**.
- **CustomLog**: registro de Apache. **`/var/log/httpd/access_log`**.
- **Timeout**: período durante el cual el servidor espera emisiones/recepciones en el transcurso de una comunicación. Por omisión, 300 segundos.
- **KeepAlive**: define si el servidor puede ejecutar más de una petición por conexión. Está en off por defecto, pero si se pasa a on, Apache puede generar rápidamente unos procesos hijos para aligerar el exceso de carga.
- **MaxKeepAliveRequests**: número máximo de peticiones por conexión persistente. Un valor elevado puede aumentar el rendimiento del servidor. 100 por defecto.
- **KeepAliveTimeout**: período durante el cual el servidor (en general un proceso hijo) espera después de haber servido una petición. Por defecto, 15 segundos. Después de 15 segundos, se recibirá la petición por el servidor con un Timeout.
- **StartServers**: número de servidores creados en el momento del inicio. Por defecto, 8. Ya que Apache gestiona de manera dinámica el número de servidores hijos, este parámetro tiene poca importancia porque el número va a bajar o a aumentar rápidamente.

5. Gestión del rendimiento

- **MaxRequestPerChild**: número de peticiones que se pueden ejecutar con un proceso hijo antes de pararse. Por defecto, 4000. Eso permite una ocupación de la memoria más reducida, liberándola más rápidamente.
- **MaxClients**: límite del número total de peticiones que se pueden tratar simultáneamente. Por defecto, 150. El valor limita el riesgo de saturación del servidor.
- **MinSpareServers / MaxSpareServers**: según la carga de la máquina, Apache puede iniciar otros procesos servidores para adaptarse a la carga actual. Por defecto, son de 5 y 20. Estos dos valores determinan los números límite autorizados. Si un servidor está poco cargado con 15 procesos, Apache suprimirá algunos, pero guardará siempre al menos 5. Si el servidor está cargado con 10 procesos, Apache creará unos adicionales, hasta 20, para situaciones de alta "concurrencia".
- **MinSpareThreads / MaxSpareThreads**: cada servidor hijo puede aceptar un determinado número de peticiones simultáneamente. Por eso utiliza los threads. Se fijan estos dos valores por defecto a 20 y 75. El mecanismo funciona como anteriormente.
- **ThreadsPerChild**: número de threads por defecto en el momento del inicio de un servidor hijo. Por defecto, 25.

6. Los directorios, alias y ubicaciones

a. Directory

Las etiquetas **<Directory ruta>** y **</Directory>** permiten agrupar directivas que no sólo se aplicarán a la ruta (y a sus subdirectorios) de los datos. Se aconseja encarecidamente la directiva**Options**.

```
<Directory /var/www/html/images>
    Options +Indexes +FollowSymLinks
    DirectoryIndex index.php index.html
    Order allow, deny
    Allow from All
</Directory>

<Directory /var/www/html/cgi-bin>
    Options +ExecCGI
</Directory>
```

La directiva **Options** acepta los valores siguientes precedidos de + o - y separados por espacios:

- **All**: todas las opciones salvo MultiViews;
- **Indexes**: si el directorio no contiene archivos HTML por defecto (vea DirectoryIndex), se muestra el contenido del directorio en forma de listing;
- **ExecCGI**: se autoriza la ejecución de scripts CGI;
- **FollowSymLinks**: el servidor sigue los vínculos simbólicos.

La directiva **DirectoryIndex** especifica los archivos html o cgi por defecto durante la carga de una URL.

```
DirectoryIndex index.php index.html
```

Durante la carga, sin precisar el nombre del archivo html, el servidor intentará cargar index.php. Si no está, entonces index.html. En el caso contrario, es la opción **Indexes** la que determina si el contenido debe mostrarse en forma de directorio.

Solo en la versión 2.2 la directiva **Allow** indica qué clientes tienen autorización para acceder al directorio. Puede ser **all**, un dominio, una IP, una IP truncada (subred), un par red/subred, etc. La directiva **Deny** prohíbe el acceso y se utiliza de la misma manera. Se determina el orden por la directiva **Order**.

A partir de la versión 2.4 las directivas Order, Allow y Deny son remplazadas por **Require**. Las líneas asociadas del ejemplo son remplazadas por:

```
Require all granted
```

b. Alias

La directiva **Alias** permite crear un atajo entre el árbol lógico del sitio Web y una ruta del sistema de archivos.

```
Alias /help "/usr/share/doc/html"
```

En este caso, la URL http://www.misitio.org/help no buscará en el directorio /var/www/html/help, sino en /usr/share/doc/html.

A diferencia de las etiquetas **<Directory>**, las etiquetas **<Location>** y **</Location>** permiten aplicar directivas basadas en la URL (y no los directorios).

```
<Location /help>
    Options +All -FollowSymLinks
```

```
    Order deny, allow
    Deny from all
    Allow from .midominio.org
       # En 2.4 las tres líneas son remplazadas por:
       # Require host midominio.org
</Location>
```

7. Hosts virtuales

Un servidor Apache es capaz de gestionar varios sitios Web en un mismo servidor. Existen varios métodos. El primero se basa en los nombres (varios sitios Web para un servidor); el otro, en las direcciones IP (una dirección IP para cada sitio web). Vamos a considerar la primera versión.

La directiva **NameVirtualHost** especifica la dirección IP en la cual el servidor va a recibir las peticiones de acceso a los anfitriones virtuales.

Las etiquetas **<VirtualHost>** y **</Virtualhost>** permiten definir un anfitrión virtual.

```
NameVirtualHost 192.168.1.3

<VirtualHost 192.168.1.3>
    ServerName      www2.midominio.org
    ServerAdmin     webmaster@www2.midominio.org
    DocumentRoot    /var/www/www2.midominio.org/
    ErrorLog        logs/www2_error_log
    CustomLog       logs/www2_access_log
</VirtualHost>
```

Vuelva a iniciar Apache. Con un navegador (p. ej.: Firefox) se verifica si nuestro anfitrión virtual contesta con la URL http://www2.midominio.org (esta dirección debe declararse en /etc/hosts o bien ser conocida por el servidor de nombres y apuntar al servidor correcto).

Observe, sin embargo, que si pasa por http://www.midominio.org ¡ya no obtiene el sitio por defecto! En efecto, cuando se declaran hosts virtuales, el primero de la lista se convierte en el anfitrión por defecto y prioritario.

Cuando se accede a un servidor, Apache busca primero una correspondencia entre el nombre de anfitrión especificado por la URL y cada Servername de los anfitriones virtuales. Si no se encuentra ninguna correspondencia exacta, se elige por defecto el primer anfitrión virtual haciendo abstracción de los parámetros globales.

Añada un anfitrión virtual para el sitio principal.

```
<VirtualHost 192.168.1.3>
    ServerName      www.midominio.org
    ServerAdmin     webmaster@www.midominio.org
    DocumentRoot    /var/www/html
    ErrorLog        logs/error_log
    CustomLog       logs/access_log
</VirtualHost>
```

Cuidado, la regla anterior se aplica también con un nombre corto. Si inscribe http://www o http://www2 se topará siempre con el anfitrión virtual por defecto. Hay que requerir: http://www.midominio.org y http://www2.midominio.org

Se pueden colocar en un anfitrión virtual todas las directivas deseadas (o casi).

Archivos compartidos

1. NFS

a. Inicio

El sistema de archivos compartidos **NFS** (*Network File System*) o sistema de archivos de red permite compartir todo o parte de un sistema de archivos destinados a clientes NFS, a menudo otros Unix. En su versión básica, es un sistema simple y eficaz. Vamos a estudiar la versión 2, pero los ejemplos funcionan también en versión 3 (observe sin embargo, que el protocolo de transporte por defecto que pasa a TCP en lugar de UDP para algunas distribuciones).

NFS se basa en el **portmapper (portmap)**, el soporte **nfs** del núcleo y los servicios **`rpc.nfsd`** y **`rpc.mountd`**.

Para iniciar el servicio NFS, se debe ejecutar portmap y nf (verificar el estado antes).

```
# service portmap status # /etc/init.d/portmap status o rpcinfo -p
# service nfs status
# service portmap start
# service nfs start
# service nfslock start
```

En algunas distribuciones, portmap se cambia por **rpcbind**. Hay que iniciar este servicio en lugar de portmap.

Para saber si el servicio está disponible en un anfitrión remoto:

```
# rpcinfo -p host
```

b. Compartir por el lado servidor

La lista de directorios que hay que exportar se encuentra en **`/etc/exports`**. Contiene un sistema compartido por línea.

```
# Dir exportes  Autorizaciones de acceso
/               puesto1(rw) puesto2(rw,no_root_squash)
/projects       *.midominio.org(rw)
/home/joe       puesto*.midominio.org(rw)
/pub            192.168.1.0/255.255.255.0(ro)
```

Cada línea se compone de dos partes. La primera es la ruta del directorio exportado. La segunda contiene las autorizaciones de acceso.

La autorización de acceso se compone de pares huésped/permisos según el formato siguiente:

```
host(permisos)
```

Si no se define el huésped, se aplicarán los permisos a toda la red (de alcance mundial). Si no se definen los permisos, la exportación será de sólo lectura. Sobre todo, no debe ponerse espacio entre el huésped y los permisos. El host puede ser:

- un nombre de host único;
- un dominio;
- una red o subred;

- una combinación del conjunto con caracteres de sustitución (*, ?).

Los permisos pueden ser:

- **ro**: sólo lectura;
- **rw**: lectura escritura;
- **no_root_squash**: el root remoto equivale al root local;
- **root_squash**: si root se conecta al directorio compartido, su uid será reemplazado por el de un usuario anónimo. Así no hay riesgo de que el usuario root de un puesto local pueda ser root en un directorio remoto;
- **all_squash**: extiende la regla precedente a todos los usuarios;
- **anonuid** / **anongid**: uid y gid para el usuario anónimo.

Para una gestión correcta de los derechos y permisos, **los usuarios de mismo nombre (login) deben tener los mismos UID y GID en el servidor y el cliente**. NFS se basa, en efecto, en estos valores para garantizar la seguridad de los datos del directorio compartido. El nombre de login solo no es suficiente. En el ejemplo siguiente, el usuario **joe** está autorizado a acceder al directorio compartido **/home/joe** (se supone que es su directorio personal) en todos los puestos del dominio. Se debe declarar al usuario joe de la misma manera (mismo UID) en el servidor y en todos los puestos. Por eso se utiliza a menudo NIS con NFS.

El comando **exportfs** permite controlar los directorios compartidos.

- **`exportfs -r`**: refresca la lista de directorios compartidos después de la modificación de /etc/exports;
- **`exportfs -v`**: lista de los directorios compartidos;
- **`exportfs -a`**: exporta (o vuelve a cargar) todos los directorios compartidos de /etc/exports o un directorio compartido dado;
- **`exportfs -u`**: detiene la compartición dada. `-a` para todos.

El comando **`showmount`** muestra los directorios compartidos de un anfitrión dado.

```
showmount -e host
```

c. Montaje por el lado cliente

El soporte NFS está incluido en forma de módulo del núcleo. Se carga automáticamente cuando se utiliza un acceso NFS.

En **`/etc/fstab`** observe las modificaciones:

```
server1:/pub /mnt/pub nfs defaults 0 0
```

La ruta del directorio compartido con la forma **servidor:ruta** ha reemplazado al periférico. El sistema de archivos es **nfs**. Es lo mismo con el comando **mount**:

```
mount -t nfs servidor1:/pub /mnt/pub
```

Si se definen los montajes NFS en /etc/fstab, mount `-a` no los va a montar. Un servicio suele estar presente en cada distribución para montarlos y desmontarlos en las paradas/reinicios. En Red Hat, el servicio /etc/rc.d/init.d/netfs los montará de manera automática durante el inicio.

NFS dispone de opciones de montaje específicas:

- **nolock**: opción de compatibilidad con antiguos servidores NFS;

- **intr**: interrumpe una petición NFS si el servidor no contesta;
- **hard**: bloquea los procesos que intentan acceder a un directorio compartido inaccesible;
- **soft**: un proceso devolverá un error en caso de acceso infructuoso;
- **rsize=8192, wsize=8192**: tamaño de los bloques de lectura/escritura en el servidor. Una escritura de 8 kB es más rápida que 8 escrituras de 1 kB.

¡Atención al tipo de montaje NFS! Durante mucho tiempo, incluidas algunas distribuciones basadas en el núcleo 2.6, los montajes NFS utilizaban el protocolo UDP. Sin embargo, esta opción de montaje se ha modificado y ahora se utiliza TCP. En UDP, cuando una transmisión fallaba, saltaba un timeout. Más allá de un valor determinado, significa que se tendrá que retransmitir un paquete RPC completo con cada error de transmisión, que es bastante pesado.

En TCP, el control de transmisión se deja en manos del propio protocolo para los paquetes más pequeños. Ahora bien, esto puede provocar efectos secundarios importantes cuando el servidor está trabajando con demasiada carga o cuando la entrada/salida es demasiado grande en un montaje, ya que el valor de timeout por defecto no ha cambiado. Entonces, puede ser que suceda lo siguiente: un archivo se escribe en el montaje NFS; por una razón cualquiera, el paquete TCP no llega en el tiempo previsto (por unas décimas de segundo) y NFS solicita la retransmisión de los datos sin esperar el timeout TCP, lo que provoca un conflicto, una corrupción de datos y la aparición de mensajes de error.

```
nfs: server not responding, timed out
```

o incluso:

```
touch: cannot touch `/home/seb/test': Input/output error
```

Hay que modificar la opción de montaje dejando TCP pero modificando el timeout NFS, expresado en décimas de segundo, para hacerlo corresponder al de TCP. Utilice un valor de 600, es decir, 60 segundos. En todas las últimas versiones de NFS para Linux, se ha convertido en el valor predeterminado, pero para estar seguro de que no se encontrará con este problema, utilice el comando como en el ejemplo siguiente:

```
mount -o timeo=600,retrans=2, tcp -t nfs servidor1:/pub /mnt/pub
```

2. FTP

El servidor **FTP** (*File Transfer Protocol*) más habitual es **vsftpd** (*Very Secure FTP Daemon*). Tiene la ventaja de ser muy pequeño, eficaz y rápido y al mismo tiempo muy configurable (sin embargo, menos que Proftpd u otros). Es adecuado para la casi totalidad de las situaciones. Se trata de un servicio que se puede iniciar tanto por **xinetd** como (en las últimas versiones de las distribuciones) a título de servicio individual.

Hay dos niveles de seguridad a disposición del usuario:

- **Anonyme**: todo el mundo se puede conectar al servidor FTP como usuario **ftp** o **anonymous**. El entorno FTP está chrooteado.
- **Usuario**: los usuarios que existen en el servidor pueden conectarse con su contraseña y tienen un acceso completo a sus datos en su directorio personal.

Dado que los usuarios anónimos se consideran como el usuario ftp, es el directorio personal de esta cuenta el que constituye la raíz del ftp.

El archivo de configuración se encuentra en /etc/vsftpd/vsftpd.conf.

La raíz del ftp por defecto está en /var/ftp.

El script de ejecución es /etc/init.d/vsftpd (servicio vsftpd start).

Para activar o no el acceso anónimo, se modifica el archivo de configuración. En este caso, el usuario se puede conectar como anonymous o ftp. En todos los casos, se reconocerá como usuario ”ftp” del servidor una vez conectado:

```
anonymous_enable=YES/NO
```

Para activar o no el envío de archivos en el servidor por anónimos. En este caso, la autorización de escritura en un directorio es función de los derechos del directorio en el servidor (en particular si el usuario ftp tiene el derecho de escribir o no en un directorio):

```
anon_upload_enable=YES/NO
```

Puede prohibir la conexión a unos usuarios colocando sus nombres en /etc/vsftpd.ftpusers.

Puede añadir usuarios en /etc/vsftpd.user_list si **userlist_enable=YES**. En este caso, es el valor de **userlist_deny (YES/NO)** el que determinará si el archivo contiene los usuarios prohibidos o autorizados.

Se puede crear en cada directorio del servidor un archivo **.message**. En ese caso, su contenido se visualizará en el momento del acceso al directorio.

Compartir Windows con Samba

1. Presentación

Samba es un conjunto de servidores que implementan los protocolos SMB/CIFS y NetBIOS/WINS para Unix. Su utilización más famosa es la de compartir recursos entre Windows y Unix, pero funciona perfectamente entre dos Unix. Una **compartición** se llama también **servicio**. Samba se compone de dos servicios:

- **smbd**: servidor SMB/CIFS.
  - Autenticación y autorización.
  - Archivos e impresoras compartidos.
- **nmbd**: servidor de nombres NetBIOS.
  - Recorrido de los recursos.
  - Servidor WINS.

Un tercer servicio, **winbindd**, permite utilizar las cuentas de usuario de un dominio Microsoft. Las últimas versiones de Samba (3 y siguientes) permiten conectarse también a Active Directory.

Las funciones principales de Samba son:

- Autenticación de los usuarios.
- Archivos e impresoras compartidos.
- Recorrido de los recursos compartidos de la red.
- Resolución de nombres (independiente de DNS) Nombre Netbios IP o viceversa.

2. Configuración

La configuración de Samba se encuentra en /etc/samba/smb.conf. Puede comprobar su sintaxis con la herramienta **testparm**.

El archivo smb.conf retoma la sintaxis de los archivos de configuración de Windows de tipo **ini** con secciones delimitadas por corchetes **[ ]**.

Por defecto hay tres secciones:

- **[global]**: ajustes genéricos y globales del servidor, nombre, comentarios, método de autenticación, ajustes por defecto, etc.
- **[homes]**: compartición de los directorios personales de los usuarios.
- **[printers]**: compartición de las impresoras.

Los parámetros tienen la forma:

```
nombre = valor
```

Los comentarios comienzan por un punto y coma **;** o una almohadilla **#**.

```
[global]
workgroup = MYGROUP
netbios name = puestoN
security = share
```

- **workgroup**: nombre de grupo/dominio de trabajo;
- **netbios name**: nombre netbios de la máquina;
- **security**: método de autenticación (vea más adelante).

3. Archivos compartidos

Cada partición debe disponer de su propia sección en smb.conf. Veamos cómo compartir el directorio **`/opt/compartición1`** con el nombre de servicio **compartición1**:

```
[compartición1]
comment = Directorio compartido 1
path = /opt/compartición1
browseable = yes
public = no
writable = yes
printable = no
group = compartición
```

Veamos una descripción de estos valores y de algunos otros posibles.

- **compartición1**: el nombre de la partición tal y como aparece en el "directorio de recursos compartidos".
- **comment**: el comentario tal y como aparece al lado del nombre del recurso compartido.
- **path**: la ruta al recurso.
- **public**: el recurso está a disposición del invitado (guest).
- **browseable**: la compartición aparece en el "directorio de recursos compartidos".
- **writable**: el recurso está accesible en modo lectura y escritura.
- **printable**: la impresora está compartida.
- **group**: nombre del grupo por defecto para la conexión.
- **valid users**: nombre de los usuarios autorizados a acceder a este recurso.
- **read only**: el recurso está en modo de sólo lectura para todo el mundo.
- **guest ok**: ninguna contraseña es necesaria para acceder al recurso. En este caso, se utilizará la cuenta invitada por defecto.
- **guest only**: el recurso está accesible únicamente a los invitados.

Se autoriza el acceso a los recursos por defecto (ver la parte sobre los métodos de autenticación) en función de los derechos Unix. En el acceso al recurso, se requiere un nombre de usuario y una contraseña. Los derechos del directorio compartido y de su contenido determinan los derechos del usuario.

4. Impresoras compartidas

Además de la sección **[printers]**, puede añadir secciones para impresoras específicas. El parámetro**printing** de la sección **[global]** permite modificar el subsistema de impresión basado por defecto en CUPS.

```
[Escritorio150]
comment = Laserjet 2100
printer = lj2100
valid users = juanito jorgito jaimito
```

```
path = /var/spool/lj2100
public = no
writable = no
printable = yes
browseable = yes
```

printer: nombre de la impresora en Linux.

- **valid users**: nombre de los usuarios autorizados a acceder a esta compartición. Una @ delante del nombre indica un grupo de usuarios.
- **path**: ruta rdel spool de impresión.

5. Métodos de autenticación

Samba ofrece varios métodos de autenticación definidos en la sección **[global]**:

- **user**: método por defecto, el acceso al conjunto de las comparticiones de un servidor se efectúa mediante la validación de un nombre de usuario y contraseña únicos.
- **share**: método de validación de los identificadores compartición por compartición. En este caso, todos los accesos a las comparticiones, incluso públicas, necesitan identificadores.
- **domain**: utilización de un grupo de trabajo con autenticación.
- **ads**: utilización de Active Directory.

Hay disponibles otros tipos de autenticación, como un servicio directorio LDAP.

```
[global]
        workgroup = MYGROUP
        netbios name = puestoN
        security = share
```

6. Correspondencia de los nombres y contraseñas

Las contraseñas del protocolo SMB no tienen la misma forma que las contraseñas Unix/Linux. Hay que volver a crear las contraseñas para cada usuario que debe utilizar SMB con el comando**smbpasswd**. Los usuarios ya deben existir en Unix.

```
smbpasswd -a pepito
```

Los usuarios SMB están presentes en **`/etc/samba/smbpasswd`**. El comando **mksmbpasswd**puede realizar esto en batch:

```
cat /etc/passwd | mksmbpasswd > /etc/samba/smbpasswd
```

Puede establecer una tabla de correspondencia entre los nombres de usuarios de Windows y los de Unix en **`/etc/samba/smbusers`**.

```
# Unix_name = SMB_name1 SMB_name2 ...
root = administrator admin administrador
```

7. Clientes SAMBA

a. En línea

Cualquier máquina en Microsoft Windows puede acceder a los recursos compartidos Samba. Los navegadores de los entornos de escritorio **KDE** y **GNOME** aceptan la navegación en las comparticiones gracias al protocolo **smb:/** en las URL. KDE propone el equivalente a un directorio de

recursos compartidos.

La herramienta **smbclient** es un tipo de cliente FTP para el protocolo SMB. Las rutas de acceso a los recursos tienen la forma:

```
//máquina/compartición
```

Por ejemplo, para conectarse al servicio (aquí un recurso compartido) de una máquina:

```
smbclient //máquina/servicio -U login%passwd
```

Para listar los servicios propuestos por una máquina:

```
smbclient -L hostname -U login%passwd # lista de los recursos
```

b. Montaje

Monte un sistema de archivos SMB con el comando **smbmount**.

```
smbmount //máquina/compartición /mnt/mountpoint -o username=login
```

También se puede efectuar el montaje en `/etc/fstab`. Al igual que nfs, es un servicio especializado que montará y desmontará los recursos compartidos. Los últimos núcleos Linux sustituyeron **smbfs** por **cifs**:

```
//máquina/compartición /mnt/mountpoint cifs defaults,username=nobody 0 0
```

Bases de seguridad

1. Seguridad informática

Los principales objetivos de la seguridad informática conciernen a:

- **La seguridad de la conexión**: se trata de controlar que los usuarios que se conectan dispongan efectivamente de la autorización para ello y de prohibirles el acceso al sistema en caso contrario.
- **La integridad de los datos**: se trata de conseguir que los archivos y las bases de datos no estén corruptas y de mantener la coherencia entre los datos.
- **La confidencialidad de los datos**: el acceso a los datos para consulta y modificación se debe limitar únicamente a usuarios autorizados.

Dispone de varios medios:

- La autentificación de los usuarios mediante contraseña.
- Encriptar los datos.
- La seguridad física controlando el acceso de las personas a las salas informáticas, mediante circuitos físicos inviolables.
- La información relativa a los riesgos penales en los que se incurre en caso de infracción. Un "atraco" informático es un delito, no un juego.
- El control frecuente de los permisos de acceso a los archivos y bases de datos.
- El control de los ”checksum” de los archivos para asegurar su integridad.
- La copia de seguridad regular de los datos.
- El control de los principales eventos del sistema.
- La instalación de cortafuegos (*firewall*) que controlan los accesos al sistema informático desde el exterior y evitan que los usuarios accedan a servicios externos sin querer o sin necesitarlo y, de esa manera, limitar el riesgo de propagación de virus.
- La instalación de un antivirus, incluso en Linux, si el servidor trata datos desde sistemas operativos susceptibles de tener virus y hacia ellos.
- La instalación de herramientas antispams y antispywares, según el mismo principio, con el fin de evitar una intrusión y la saturación de los servidores de correo electrónico.
- Iniciar únicamente los servicios realmente útiles en el servidor y el cliente.

Algunos métodos sencillos permiten limitar los riesgos:

- Puede definir un valor de umask restrictivo (p. ej.: 077) para extender a continuación los permisos de acceso de algunos archivos.
- No debe apartarse del terminal sin desconectarse o bloquearlo (una buena broma en caso contrario consiste en utilizar el cliente de correo de la persona en cuestión para que le traigan una pizza…).
- Hay que prestar atención a las fechas de los últimos inicios de sesión logrados e infructuosos que aparecen en cada conexión.
- No permitir nunca el acceso, incluso en modo de sólo lectura, al archivo .profile.
- Nunca poner el ”.” en primera posición del PATH, y controlar sus rutas.

2. Controlar los privilegios especiales

Los privilegios especiales de ejecución (bits SUID y SGID) suelen ser causa de inseguridad en el sistema. En efecto, un usuario malintencionado, aprovechando la falta de atención o la ausencia de un compañero o un administrador que no está desconectado de su consola, puede modificar los permisos de ciertos comandos a su favor. El ejemplo más habitual es el de reescribir un shell como un programa poco usado (por ejemplo sx) y darle los privilegios SUID. Al iniciar este comando, se puede convertir en root.

Obtener el permiso de listar todos los archivos

```
# chmod u+s cat
```

Obtener un shell root

```
# cp /bin/sh /bin/sx
# chmod u+s /bin/sx
...
$ sx
# ...
```

El comando siguiente permite buscar todos los archivos que disponen de los bits SUID o SGID:

```
# find / -type f \( -perm -4000 -o -perm -2000 \)
# find / -type f \( -perm -4000 -o -perm -2000 \)
/bin/su
/bin/umount
/bin/eject
/bin/mount
/bin/ping
/bin/ping6
/sbin/unix2_chkpwd
/sbin/unix_chkpwd
/usr/bin/expiry
/usr/bin/write
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gnotski
/usr/bin/mahjongg
/usr/bin/chfn
/usr/bin/yset
/usr/bin/wall
/usr/bin/crontab
/usr/bin/v4l-conf
/usr/bin/gnomine
/usr/bin/same-gnome
/usr/bin/gnotravex
/usr/bin/gnobots2
...
```

En la lista anterior, hay un intruso: **/usr/bin/yset**, que permite modificar las configuraciones de un servidor de sonido y que no necesita para nada disponer del privilegio SUID. Hay un problema.

```
# ls -l /usr/bin/yset
-rwsr-sr-x 1 root root 604040 may 19 21:28 /usr/bin/yset
# md5sum /usr/bin/yset
04ff72010ff1cf1c14d7706159cdf8bf  /usr/bin/yset
# ls -l /bin/bash
-rwxr-xr-x 1 root root 604040 sep 22  2007 /bin/bash
# md5sum /bin/bash
04ff72010ff1cf1c14d7706159cdf8bf  /bin/bash
```

Alguien ha vuelto a escribir un shell con otro nombre.

3. Comprobar los paquetes

El capítulo Instalación de Linux y de los paquetes de software ha tratado sobre toda la gestión de los paquetes de software. Entre las diversas opciones, algunas permiten controlar la autenticidad de un paquete. El sistema de paquetería RPM contiene, además del nombre del archivo, su tipo (configuración, binario, etc.) y en algunos casos (binario) la suma de control (checksum) MD5 del archivo.

Según el ejemplo anterior, ¿cómo restaurar el archivo yset? En tres etapas:

- Encontrar el paquete de origen:

```
# rpm -qf /usr/bin/yset
yiff-2.14.5-0.pm.1
```

- Controlar el estado del paquete instalado:

```
# rpm -V yiff
SM5....T    /usr/bin/yset
```

    - S: el tamaño no es el correcto
    - M: se han modificado los permisos
    - 5: la suma de control MD5 es diferente
    - T: la fecha de modificación no es la correcta.

- Vuela a instalar el paquete de origen según las modalidades propias de su distribución.

4. Política de la contraseñas

Las contraseñas son la base de la autenticación de un usuario. Deben ser seguras. Sin embargo, suele ser la signatura pendiente, tanto en el trabajo como en casa, e incluso en Internet:

- contraseña escrita en un post-it;
- uso de un gestor de contraseña automático, a su vez sin contraseña;
- misma contraseña para todos los sitios web y software;
- contraseña nunca cambiada;
- misma contraseña o cuenta para a toda la familia/servicio;
- contraseña demasiado sencilla;
- etc.

No sirve de nada caer en la paranoia. Tiene que encontrar un término medio. Si pide a los usuarios que modifiquen su contraseña demasiado a menudo, o si es demasiado difícil, tienden a apuntarla. Si es demasiado fácil y deja pasar demasiado tiempo, ya no es seguro.

Los usuarios deben elegir una buena contraseña, evitando la sencillez o más bien lo evidente: nombres de los hijos, de la esposa, de lugares, fecha de nacimiento y, en general, todo lo que importa y que es conocido del entorno profesional o personal.

Un término medio puede ser modificar las reglas de cambio de contraseña con chage (o passwd), de forma que se establezca una duración para la validez de la contraseña de 40 días. Se han presentado los comandos de modificación de la política de gestión de las contraseñas en el capítulo Las tareas administrativas - Administración de los usuarios.

```
# chage -l bean
```

```
Minimum:        7
Maximum:        40
Warning:        10
Inactive:       5
Last Change:            abr 10, 2008
Password Expires:       may 20, 2008
Password Inactive:      may 25, 2008
Account Expires:        ene 01, 2010
```

Los módulos PAM influyen en la política de gestión de las contraseñas, obligando en algunos casos a elegir uno más o menos complejo. Aunque parezca una paradoja, una contraseña debe ser fácil de recordar por un usuario, lo que no implica forzosamente que sea fácil de piratear (por John the Ripper, por ejemplo). Existen contraseñas que se pueden recordar por medios mnemotécnicos. También puede generar contraseñas de manera automática con la herramienta **pwgen**.

```
$ pwgen
uash6She lohJo7ae Ohphab3i ouRik9ie uM4va3im Neer7Eit eib3Hauy xo9Iuy5p
ahSiW0uf AhG6wail Yai6neeh phae4ioV deeL3aip Uz5ahzaa aiV5phee Aegaiy7x
ioPh1ahn Ong6Baib Eish4rip eik9Gie1 ien3Iepe xohduj7U aiP2keov So5ovaht
Voh9oxoe ahs2Meeg Ooch5xix Phe3yiuz eeCa5ohv aig9Ai3o Go4Ateeh Hee6thei
Rai6Daeh aid8ieNg Thah6ien daphaiG0 Iefai5oh Pheife6i Poora8ah Coh5Aida
ViC7ieth hohG5sei Aa9Jeilu eopoX8Si jooh3Eif dooPhai1 chohqu1G ieNgae3o
wiCeiSi3 aej6Piev eoTha1Fu ieR2yeeb Eireili6 saiGhie2 XohRoo1a cahb2Yah
Guungah0 ube3vo0D oshol3Op Pui6agh5 Ao7baeN1 foTek9Ei aeM3lala Ene2baol
geloV9ai Weeyu2ie Uvae2Vie dei0euL7 Xee9uaza ed8Eeghu eebiu2Ka zey0Liuh
be6Ailoi eiph8Ohb Yahpahr4 aij4dahG oQu2chae Fe5eeg9c Hoosh6oh Iip8eiwe
AuPie0um Ahxai9eo Dae5oquu Ie7Viek8 pa2aew8B fohham7A fah1Oogi ieH9vee8
saeC8sha Aejeey6i Eithoow1 yi9vei0L ohC7eegh IaTh4ohn ti6Fooxe Oiche7oh
Tah9uos7 Paej2Iec chuiD8ei aicoGh5l saiKeiw2 mae9mieY Ais9oanu Mah9xej3
Zi2nacai gaiM4thi sapa1Fah kie8oZo7 Po5uuho8 thae3Aim Ohjahgh9 Weike8ra
Cah4weiZ teoji0Oo vi0hei6O Zieha3ai Keip2bie bahR7bah ahSai0Ei afoh3Thi
eeNieTh8 Zei7eth8 uV5eichi kue1Eedi sueThe0V wohChe2u Ohl1zicu Loo1soo3
yahb9uSi EelieGh1 aeMiThi1 OoFoh8wu Ieyei5ka Roph7ape uem5quuK ahQu7eec
NahSha6A kooMou0y gu1chaiJ hae2ku0Z uC3oeNgo xuSha7qu Iucai0fu uK4icewe
eP7aetig ahYai0ee Eetahfu8 yeep0oPi Veimaij3 Oht0aiPh buTh9oob ood4nieC
sah7Ahj1 koozah0J Vieb9Bit eeP9nee1 ea1SohCe Afei4ohS eikahk0W rachoG4c
```

Estas contraseñas son pseudoaleatorias. Si habla inglés (y geek/leet), estas contraseñas representan una pronunciación. Por ejemplo:

```
dooPhai1: Do you fail?
```

Puede pedir que se generen contraseñas totalmente aleatorias con una longitud dada, en este caso de 10 caracteres:

```
$ pwgen -s -1 10
ER9BAgHsZH
```

5. Prohibir las conexiones

a. /bin/false

Algunas cuentas no deben ser interactivas: se deben prohibir las conexiones desde una consola. Se pueden asignar estas cuentas a una aplicación, a un servicio, a una conexión FTP, etc., pero se debería rechazar la conexión: ¡no shell!

En las lista de los shells autorizados, uno llama la atención:

```
$ cat /etc/shells
/bin/ash
/bin/bash
```

```
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
/bin/zsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/passwd
/usr/bin/bash
/usr/bin/tcsh
/usr/bin/zsh
```

El shell **/bin/false** no es realmente un shell. Prohíbe las conexiones interactivas. Ya hemos encontrado este comando **false** en el capítulo El shell y los comandos GNU: devuelve siempre falso. En cuanto **login** intenta ejecutar el shell de conexión, se rechaza al usuario.

```
$ cat /etc/passwd|grep false
avahi:x:104:106:User for Avahi:/var/run/avahi-daemon:/bin/false
haldaemon:x:101:102:User for haldaemon:/var/run/hal:/bin/false
icecream:x:102:103:Icecream Daemon:/var/cache/icecream:/bin/false
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
messagebus:x:100:101:User for D-Bus:/var/run/dbus:/bin/false
ntp:x:74:104:NTP daemon:/var/lib/ntp:/bin/false
polkituser:x:103:105:PolicyKit:/var/run/PolicyKit:/bin/false
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
vscan:x:65:110:Vscan account:/var/spool/amavis:/bin/false
```

b. /etc/nologin

Antes de pasar por un seudoshell de conexión, los módulos PAM permiten aplicar numerosas limitaciones. Entre ellas, el módulo **pam_nologin**, que le permite prohibir la conexión de los usuarios excepto root. Este módulo monitoriza si un usuario intenta conectarse al archivo **/etc/nologin**. Es útil para tareas de mantenimiento, ya que sólo root se puede conectar.

```
$ pwd
/etc/pam.d
$ grep nologin *
login:auth       requisite      pam_nologin.so
ppp:auth         required       pam_nologin.so
sshd:auth        requisite      pam_nologin.so
```

Piense también que puede prohibir el acceso de una cuenta dada vía el módulo PAM **pam_listfile**.

c. /etc/securetty

En el mismo estilo, el archivo **/etc/securetty** contiene la lista de los terminales considerados como seguros. Para el servicio dado, se prohibirá la conexión si el terminal de la persona que intenta conectarse no está incluido. En el ejemplo siguiente, se autorizan los logins (mediante el comando login) únicamente desde los seudoterminales locales, es decir, sólo desde las consolas directamente accesibles en el ordenador mediante las combinaciones [Alt][F1] a [Alt][F7].

```
$ grep securetty *
login:auth       [user_unknown=ignore success=ok ignore=ignore
auth_err=die default=bad]        pam_securetty.so
$ cat /etc/securetty
```

```
#
# This file contains the device names of tty lines (one per line,
# without leading /dev/) on which root is allowed to login.
#
tty1
tty2
tty3
tty4
tty5
tty6
```

¡No confunda securetty (Secure tty) con security!

6. Probar las contraseñas

Las herramientas crack y John the ripper intentan encriptar sus contraseñas, tanto desde un diccionario como por la fuerza bruta (unas tras otras). En el peor de los casos, las encuentran en unos segundos; en el mejor, en varios días, incluso semanas. Si esto es así, se puede considerar que la contraseña es buena.

John the ripper es muy fácil de usar. El comando es **john**. Para probar la integridad de su archivo,**/etc/shadow**:

```
# john /etc/shadow
Loaded 5 password hashes with 5 different salts (OpenBSD Blowfish [32/64])
```

En el modo por defecto, john:

- intenta una detección simple vía combinaciones corrientes relacionadas con la cuenta;
- pasa al modo diccionario con aplicación de reglas;
- luego, intenta una búsqueda incremental.

¡Una búsqueda puede tardar de unos segundos a varias semanas!

Para probar un solo usuario:

```
# john -user:seb /etc/shadow
```

Para probar los usuarios con un diccionario:

```
# john -users:seb -wordlist:/var/lib/john/wordlists/all /etc/shadow
```

Lo mismo pero probando varias reglas por palabra (inversión, mayúsculas, minúsculas, etc.):

```
# john -users:seb -wordlist:/var/lib/john/wordlists/all -rules /etc/shadow
```

Para retomar una búsqueda interrumpida ([Ctrl] **C**):

```
# john -restore
```

John ubica sus resultados en el directorio ~/.john. Generalmente sólo root debe utilizar esta herramienta:

```
# ls -l .john
total 4
-rw------- 1 root root 70 may 23 21:58 john.pot
-rw------- 1 root root 124 may 23 21:54 john.rec
```

El archivo `john.pot` contiene los resultados encontrados por John. El archivo no está vacío y esto indica que hay un problema: john ha encontrado una contraseña. El archivo **`john.rec`** contiene el estado actual de la búsqueda por si se produce una interrupción y es utilizado por el programa para reanudar la búsqueda.

En una máquina basada en Intel Core 2 duo de 64 bits a 3,4 GHz, se craqueó la contraseña de Seb (el autor) en 4 minutos y 22 segundos. Estaba (de manera voluntaria esta vez) basada en una palabra del diccionario.

Si pulsa una tecla durante la búsqueda, John muestra su estado.

Imaginemos una cuenta Enrique cuyo propietario tiene como contraseña la misma palabra que su nombre de cuenta, a saber "Enrique":

```
# john -users:enrique /etc/shadow
Created directory: /root/.john
Loaded 1 password hash (OpenBSD Blowfish [32/64])
enrique            (enrique)
guesses: 1  time: 0:00:00:00 100% (1)  c/s: 4.34  trying: enrique
```

Eso debería hacer reflexionar a los que piensan que nadie vendrá a molestarle y debería animarlos a modificar su contraseña con reglas precisas (mayúsculas, minúsculas, cifras, etc.).

7. Buscar rootkits

a. Fundamentos del rootkit

Una vez que un pirata informático haya conseguido, mediante un fallo o una contraseña demasiado sencilla, penetrar en su máquina, seguramente tratará de abrirse una puerta de entrada mayor, o más bien una puerta trasera, un **backdoor**, para así volver a utilizar su máquina con propósitos dudosos, o llevarse o almacenar datos (de esta manera, algunos PC se han encontrado con miles de archivos de audio o de video, o han sido utilizados como plataforma de ataque (PC zombi) por personas poco escrupulosas).

Lo ideal para el pirata es poder acaparar los derechos de root. Es su objetivo: ser el dueño total de su máquina. Para ello, no siempre le hace falta "teclear" directamente en esta cuenta. En Linux, es habitual (e incluso aconsejable) conectarse como simple usuario y luego cambiar a root mientras se efectúan las operaciones necesarias. Ahora bien, para pasar a root, puede utilizar **su**o **sudo**.

Si, mediante el servicio ftp mal configurado o mediante ssh (y luego scp), el pirata intenta conectarse a una cuenta cuya contraseña es obvia (como en el ejemplo de la cuenta enrique), y consigue desplegar en el sistema un script llamado "su" y modifica el PATH por defecto para poner en él la ruta al script, entonces, es pan comido.

```
$ pwd
/home/seb
$ cat su
#!/bin/bash
echo -e "Contraseña :\c"
read -s password
echo "$@ $password" > /tmp/fic
echo
echo "su: Contraseña incorrecta."
/bin/su $@
$ chmod +x su
$ export PATH=$HOME:$PATH
$ su - root
Contraseña: ===> FALSO SU
su: Contraseña incorrecta. ===> FALSO SU
Contraseña: ===> VERDADERO SU
```

Sólo falta mostrar el contenido del archivo para obtener la contraseña:

```
$ cat /tmp/arc
- root azerty
```

El método, simplista pero eficaz, no es imparable ni mucho menos: su sólo pide la contraseña una vez, y a no ser que se sea muy despistado, el engaño salta a la vista. Al hecho de ubicar scripts, modificar el entorno, sustituir un archivo por otro de modo que se obtenga un acceso privilegiado a una máquina, se le denomina instalar un **rootkit**. Una vez en su sitio, este último garantiza, mientras no se detecte, un acceso root a la máquina.

b. chkrootkit

La herramienta **chkrootkit** es una herramienta sencilla que permite buscar la presencia de los rootkits más conocidos y habituales. Sólo es eficaz si se actualiza y se ejecuta regularmente; no sustituye a los controles ya nombrados anteriormente.

```
# chkrootkit
ROOTDIR is '/'
Checking 'amd'... not found
Checking 'basename'... not infected
Checking 'biff'... not found
Checking 'chfn'... not infected
Checking 'chsh'... not infected
Checking 'cron'... not infected
Checking 'crontab'... not infected
...
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while...
...
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmind/IIS Worm... nothing found
...
Searching for ENYELKM rootkit default files... nothing found
Searching for anomalies in shell history files... Warning: '' is
linked to another file
Checking 'asp'... not infected
Checking 'bindshell'... not infected
Checking 'lkm'... chkproc: nothing detected
Checking 'rexedcs'... not found
Checking 'sniffer'... eth0: not promisc and no PF_PACKET sockets
vmnet8: not promisc and no PF_PACKET sockets
vmnet1: not promisc and no PF_PACKET sockets
Checking 'w55808'... not infected
...
```

8. Los virus

Ya han aparecido los primeros virus en Unix. Aunque se suele dotar de mayor seguridad al sistema y aunque los virus en las plataformas Unix y Linux (incluyendo Mac OS X) son casi inexistentes, en el caso más o menos probable de que un virus penetre y comprometa la seguridad de su máquina, las de otros o la de toda la red, tiene la responsabilidad de erradicarlo.

Si su máquina es un servidor, en particular de correo electrónico o de archivos en una red que contiene máquinas con Windows muy expuestas, no debe servir de vector indirecto de propagación. Tiene que eliminar la amenaza.

Existe varios antivirus en Linux, algunos comerciales (gratuitos o no), otros libres. El antivirus Clam (ClamAV) es libre y gratuito. También es uno de los que ofrece mejor rendimiento. Está disponible en la dirección http://www.clamav.net/. Se actualiza cada día.

Freshclam permite actualizar las bases de firmas, ubicadas en **/var/lib/clamav**:

```
# pwd
/var/lib/clamav
# freshclam
ClamAV update process started at Wed Jul 16 15:12:16 2014
Downloading main.cvd [100%]
main.cvd updated (version: 55, sigs: 2424225, f-level: 60, builder: neo)
daily.cvd is up to date (version: 19193, sigs: 1082296, f-level: 63, builder: neo)
bytecode.cvd is up to date (version: 242, sigs: 46, f-level: 63, builder: dgoddard)
Database updated (3506567 signatures) from db.local.clamav.net (IP: 128.177.8.248)

# ll
total 89944
drwxr-xr-x  2 clamav clamav     4096 jul. 16 15:16 ./
drwxr-xr-x 47 root   root       4096 jul. 16 15:11 ../
-rw-r--r--  1 clamav clamav    74230 jul. 16 15:16 bytecode.cvd
-rw-r--r--  1 clamav clamav 27287327 jul. 16 15:16 daily.cvd
-rw-r--r--  1 clamav clamav 64720632 jul. 16 15:16 main.cvd
-rw-------  1 clamav clamav      104 jul. 16 15:16 mirrors.dat
```

Clamscan permite buscar los posibles virus:

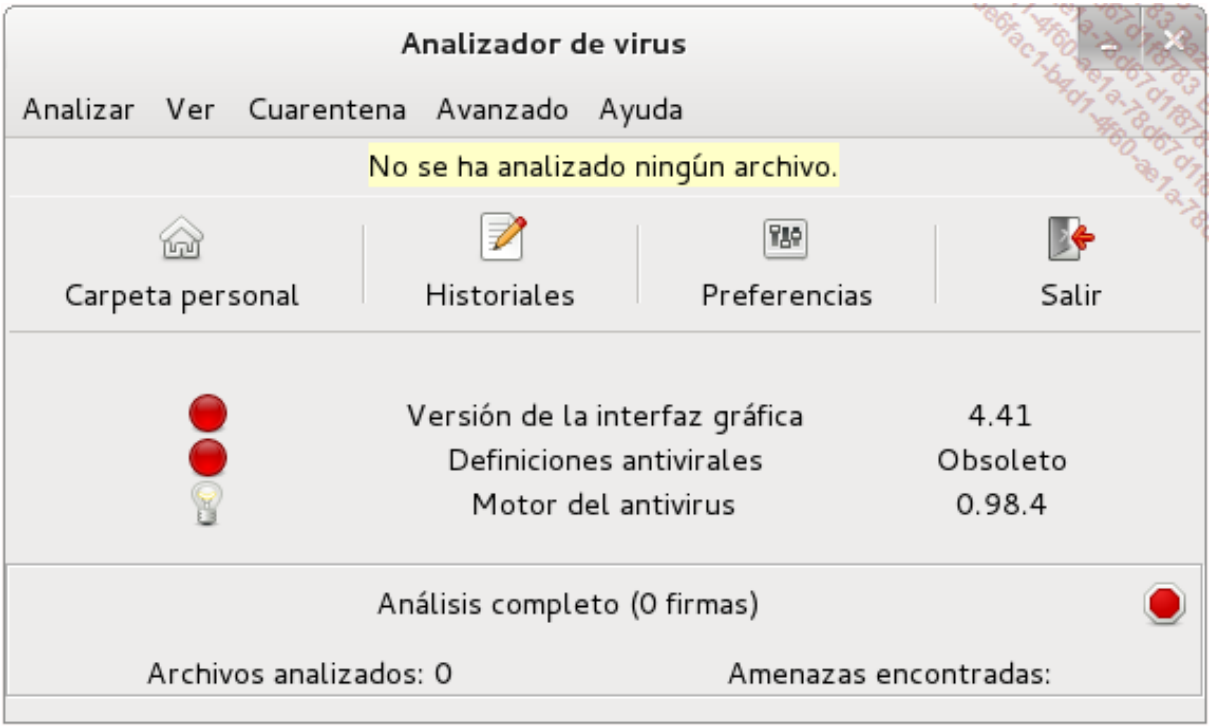
```
# clamscan -v /usr/local/bin
Scanning /usr/local/bin/supervisord
/usr/local/bin/supervisord: OK
Scanning /usr/local/bin/pip2
/usr/local/bin/pip2: OK
Scanning /usr/local/bin/pyrsa-encrypt
/usr/local/bin/pyrsa-encrypt: OK
Scanning /usr/local/bin/pyrsa-decrypt
/usr/local/bin/pyrsa-decrypt: OK
...
----------- SCAN SUMMARY -----------
Known viruses: 3501047
Engine version: 0.98.1
Scanned directories: 1
Scanned files: 30
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 9.563 sec (0 m 9 s)
```

A continuación, la misma prueba con un virus falso bajo tres formas: binario, comprimido gzip y comprimido bzip2. Si se detecta un virus, se mueve el archivo correspondiente a **/home/seb/VIRUS**:

```
$ clamscan -v -r --move=/home/seb/VIRUS /home/seb/bin
Scanning /home/seb/bin/eicarcom2.zip
/home/seb/bin/eicarcom2.zip: Eicar-Test-Signature FOUND
```

```
/home/seb/bin/eicarcom2.zip: moved to '/home/seb/VIRUS//eicarcom2.zip'
Scanning /home/seb/bin/eicar_com.zip
/home/seb/bin/eicar_com.zip: Eicar-Test-Signature FOUND
/home/seb/bin/eicar_com.zip: moved to '/home/seb/VIRUS//eicar_com.zip'
Scanning /home/seb/bin/eicar.com
/home/seb/bin/eicar.com: Eicar-Test-Signature FOUND
/home/seb/bin/eicar.com: moved to '/home/seb/VIRUS//eicar.com'

----------- SCAN SUMMARY -----------
Known viruses: 1026485
Engine version: 0.97
Scanned directories: 1
Scanned files: 3
Infected files: 3
Data scanned: 0.00 MB
Time: 1.216 sec (0 m 1 s)
```

Se puede iniciar Clamav como servicio. En este caso, se puede configurar la búsqueda de virus en estructuras concretas, lo que permite la existencia de herramientas como clamtk o la integración de funciones en gestores de archivos (como Nautilus en Gnome).



Carpeta personal analizada por ClamTk

9. Los límites del usuario

El campo de acción de los PAM es más amplio que la simple conexión, ya que gestiona también el entorno del usuario. Incluso antes de ver el módulo en cuestión, el comando **ulimit** permite actuar en el entorno del shell y de los procesos que controla. El parámetro `-a` muestra las opciones controladas por **ulimit**:

```
$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 16380
max locked memory       (kbytes, -l) 32
```

```
max memory size          (kbytes, -m) 1753125
open files                       (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority               (-r) 0
stack size               (kbytes, -s) 8192
cpu time                (seconds, -t) unlimited
max user processes               (-u) 16380
virtual memory           (kbytes, -v) 3333600
file locks                       (-x) unlimited
```

Las líneas en negrita merecen su atención.

- **max memory size**: el tamaño de memoria máximo que puede ocupar el usuario;
- **open files**: el número máximo de descriptores de archivos; por lo tanto, el número máximo de archivos que se pueden abrir;
- **max user processes**: el número máximo de procesos que puede iniciar un usuario.

Se pueden cambiar estos valores según ciertos límites impuestos por el administrador. Existen límites soft (suaves o bajos) que son los valores por defecto devueltos por ulimit, y límites hard (duros, altos), que no se pueden superar.

Para pasar el número máximo de archivos abiertos a 2048:

```
$ ulimit -n 2048
$ ulimit -n
2048
```

El administrador root puede controlar los valores por defecto gracias al archivo **/etc/security/limits.conf**.

```
$ grep seb /etc/security/limits.conf
seb             hard    nproc          32768
seb             soft    nofile          1024
seb             hard    nofile          4096
```

En este ejemplo, el usuario Seb está limitado a un máximo de 32.768 procesos, puede abrir por defecto 1.024 archivos, pero puede montar 4.096 mediante una acción ulimit por su parte.

10. Los derechos SUDO

a. Proporcionar privilegios extendidos

El comando **sudo** permite asignar el derecho a ejecutar comandos de administrador a uno o varios usuarios, en una o varias máquinas. En la práctica, para que un usuario pueda ejecutar un comando que, en principio, sólo puede ejecutar root, debe añadirse un derecho sudo a este usuario para este comando.

El archivo de configuración de sudo es **/etc/sudoers**. Es posible editarlo a mano o con el comando **visudo**. Este último comando comprueba la sintaxis del archivo en el momento de guardar.

La clásica sintaxis de una línea sudo es la siguiente:

```
user    máquina = (user2) comando
```

- **user**: el usuario (o alias) al cual se aplica la regla.
- **máquina**: la máquina (o el alias) en la cual se aplica la regla.

- **user2**: la cuenta con la cual el usuario ejecutará el comando.
- **comando**: el comando que se va a ejecutar.

Por ejemplo, la línea siguiente va a autorizar al usuario Seb a ejecutar el comando fsck y sus parámetros con los derechos root sobre cualquier máquina (donde esté presente esta regla):

```
seb         ALL = /sbin/fsck
```

Para utilizar **fsck**, Seb debe usar el comando **sudo** como a continuación:

```
seb@slyserver:~/handbrake> sudo /sbin/fsck
seb's password:
fsck 1.41.1 (01-Sep-2008)
e2fsck 1.41.1 (01-Sep-2008)
...
```

Por defecto, se pide la contraseña del usuario Seb antes de proseguir. El usuario puede obtener la lista de sus derechos sudo:

```
seb@slyserver:~/handbrake> sudo -l
User seb may run the following commands on this host:
    (root) /sbin/fsck
```

Por defecto, se autentica el usuario una primera vez; luego, no se vuelve a pedir su contraseña mientras continúe en la misma sesión (mientras no cierre su consola o su entorno) o durante la duración indicada por la opción **timestamp timeout** del archivo de configuración.

La ventaja de **sudo**, además de asignar derechos puntuales a un grupo de personas determinadas, es la trazabilidad. Se transmiten los mensajes de **sudo** a **syslog**, que los puede volver a dirigir a un archivo. Puede ser `/var/log/messages`:

```
May  8 14:39:08 slyserver sudo:      seb
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=/sbin/fsck
May  8 14:40:14 slyserver sudo:      seb
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=list
```

Se puede establecer el registro de destino mediante syslog.conf, rsyslog.conf o syslog_ng.conf, según su distribución.

b. Sintaxis de /etc/sudoers

El ejemplo anterior es deliberadamente limitado. Es posible:

- crear grupos de usuarios,
- crear grupos de máquinas,
- crear grupos de comandos,
- forzar o no el uso de una contraseña,
- forzar la ejecución de un comando bajo un usuario distinto de root.

Los grupos se llaman alias.

Para crear alias de usuarios, utilice esta sintaxis:

```
User_Alias ADMINS = seb, esteban, enrique
```

Si todos los administradores deben poder utilizar el comando **fsck**, la línea se convierte en:

```
ADMINS ALL= /sbin/fsck
```

Si los administradores sólo pueden ejecutar estos comandos en determinadas máquinas, cree un alias de máquinas:

```
Host_Alias SERVERS= slyserver, eeepc
```

La línea sudo se convierte en:

```
ADMINS SERVERS= /sbin/fsck
```

Se pueden añadir varios comandos sucesivamente, con o sin parámetros. Si los ADMINS deben ejecutar también el comando /sbin/dumpe2fs, la línea se convierte en esto:

```
ADMINS SERVERS=/sbin/fsck, /sbin/dumpe2fs
```

o, para saltar a la línea siguiente, ponga una barra oblicua "\" en el final de línea:

```
ADMINS  SERVERS=/sbin/fsck, \
            /sbin/dumpe2fs
```

Puede crear alias de comandos para agruparlos:

```
Cmnd_Alias ADMCMD=/sbin/fsck, /sbin/dume2fs
```

La línea sudo se convierte en:

```
ADMINS SERVERS=ADMCMD
```

Para evitar la introducción de la contraseña de un usuario, escriba NOPASSWD como a continuación:

```
ADMINS SERVERS=NOPASSWD: ADMCMD
```

Los usuarios del alias ADMINS ya no tendrán que teclear contraseña para introducir los comandos del alias ADMCMD.

También puede forzar la utilización de una contraseña con PASSWD. Añada así el comando mkfs:

```
ADMINS SERVERS=NOPASSWD: ADMCMD, PASSWD:/sbin/mkfs
```

Para permitir que los ADMINS inicien un comando con la identidad de otro usuario distinto de root, sitúe el nombre del usuario (o alias) entre paréntesis, como a continuación:

```
ADMINS ALL=(esteban) PASSWD: /sbin/service
```

ADMINS podrá ejecutar /sbin/service como Esteban introduciendo la contraseña de éste.

El resultado del `sudo -l` asociado es el siguiente:

```
seb@slyserver:~> sudo -l
User seb may run the following commands on this host:
    (root) NOPASSWD: /sbin/fsck, /sbin/dumpe2fs
    (root) /sbin/mkfs
    (esteban) /sbin/service
```

Para iniciar el comando bajo otro usuario, utilice el parámetro `-u` de sudo:

```
seb@slyserver:~> sudo -u esteban /sbin/service
```

Puede precisar varios usuarios, pero también crear alias al igual que para el resto:

```
Runas_Alias LISTUSR=seb,esteban
```

Puede utilizar este alias en la línea sudo:

```
ADMINS ALL=(LISTUSR) PASSWD: /sbin/service
```

Puede utilizar el alias ALL: define el conjunto de usuarios, máquinas y comandos, según la posición donde está escrito. La línea siguiente significa que los ADMINS tienen permiso para ejecutar todos los comandos, en todas las máquinas, sin contraseña:

```
ADMINS ALL=NOPASSWD: ALL
```

El signo de exclamación permite excluir a determinados usuarios, comandos, máquinas, de un alias o de una lista. La línea siguiente da todos los derechos sobre todos los programas sin contraseña a los ADMINS, salvo sobre `/sbin/mkfs`, que requiere una contraseña, y `/sbin/resize2fs`, que está prohibido:

```
ADMINS SERVERS=NOPASSWD: ALL, !/sbin/resize2fs, PASSWD:/sbin/mkfs
```

Si un ADMINS intenta iniciar `resize2fs`, obtiene un error:

```
seb@slyserver:~> sudo /sbin/resize2fs
Sorry, user seb is not allowed to execute '/sbin/resize2fs' as root
on slyserver.
```

La regla siguiente, muy peligrosa, autoriza a todo el mundo a ejecutar lo que quiera, en cualquier sitio y sin contraseña, como cualquier usuario:

```
ALL ALL=(ALL) NOPASSWD: ALL
```

A continuación viene una línea que se utiliza con frecuencia para autorizar a un usuario a hacerlo todo, salvo iniciar shells o el comando **su**, con la condición de que los alias SU y SHELLS estén completos:

```
seb ALL = ALL, !SU, !SHELLS
```

Esta línea no garantiza que Seb pueda obtener derechos por medios indirectos, por ejemplo, si se han vuelto a nombrar todos los comandos, o por rebotes de comandos sucesivos.

11. Auditoría más completa

Para efectuar la auditoría de un sistema, puede utilizar, además del acceso a los rastros e historiales, y de los comandos ya listados, los productos libres o gratuitos como tripwire, que comprueba la integridad del sistema, Nessus que vigila la seguridad del sistema, y Crack, que detecta las contraseñas erróneas.

12. Los boletines de seguridad

a. CERT: Computer Emergency Response Team

Historia

El primer virus capaz de replicarse en Internet (se llamaba Arpanet en aquella época) fue obra de un estudiante de la universidad de Cornell. Desarrollado y lanzado a finales de 1998 sin intención de perjudicar, este programa se propagaba y replicaba sólo aprovechando fallos de seguridad de Unix y sus servicios. Este programa saturó rápidamente la red y las máquinas que había alcanzado,

paralizando Internet, compuesta entonces por 60.000 ordenadores, durante varios días. Sólo se había afectado al 4 % de las máquinas.

Para erradicar este virus se necesitó tiempo y muchos medios aportados por el MIT, Berkeley, etc. Hubo que estudiar su funcionamiento para entender cómo se había comportado y sólo entonces se pudieron corregir los agujeros de seguridad de los sistemas y servidores. Se aplicaron parches correctivos. El DARPA, iniciador del proyecto Arpanet (luego Internet) introdujo más tarde una nueva estructura llamada el CERT (CERT/CC, *CERT Coordination Center*), encargada de analizar las amenazas futuras y la vulnerabilidad de los sistemas.

Internet representa decenas de millones de máquinas interconectadas, con sistemas operativos y diferentes servicios. El CERT (http://www.cert.org/), cuya sede se encuentra a día de hoy en la universidad de Canergie Mellon, sigue estudiando las posibles vulnerabilidades en Internet, incluso a largo plazo, para obtener la mejor seguridad posible.

Papel del CERT

Las funciones del CERT son las siguientes:

- centralización y análisis de las peticiones de asistencia después de los incidentes de seguridad (ataques) sobre las redes y los sistemas de información: recepción de las peticiones, análisis de los síntomas y correlación de los incidentes;
- tratamiento de las alertas y reacción a los ataques informáticos: análisis técnico, intercambio de información con otros CERT, contribución a estudios técnicos específicos;
- establecimiento y mantenimiento de una base de datos de las vulnerabilidades;
- prevención gracias a la publicación de información sobre las precauciones que es preciso tomar para minimizar los riesgos de incidente, o sus consecuencias;
- posible coordinación con otras entidades: centros de competencia de redes, operadores y proveedores de acceso a Internet, CERT nacionales e internacionales.

Boletines del CERT

Existen varios tipos de comités CERT: por país, por industria, etc. Se emiten boletines independientes, pero vinculados entre ellos. A continuación indicamos dos sitios donde encontrar alertas: https://www.ccn-cert.cni.es

Y sobre todo, http://www.us-cert.gov/

El fallo de seguridad Heartbleed en la sede del CERT

Un ejemplo de alerta es un fallo de seguridad de la librería OpenSSL en Debian *Debian/Ubuntu OpenSSL Random Number Generator Vulnerability*, Referencia TA08-137A. Las claves aleatorias generadas desde la versión OpenSSL de Debian y Ubuntu no lo son realmente, lo que limita enormemente la seguridad de estas claves y conlleva un problema, ya que la actualización de la librería no es suficiente; también hay que generar de nuevo las claves existentes... Un ejemplo muy reciente es el fallo de seguridad heartbleed, que atañe a OpenSSL, referencia TA14-098A.

b. SecurityFocus

SecurityFocus (anteriormente Bugtraq) es una lista de difusión electrónica (e-mailing) en línea desde 1993 que agrupa discusiones sobre vulnerabilidades, seguridad, anuncios y medios para detectar fallos y cómo corregirlos. Originalmente la lista nació de la necesidad de erradicar dos problemas:

- los muchos fracasos del CERT a la hora de prevenir problemas;
- la desidia de muchos editores y fabricantes, que no suministraban las actualizaciones de seguridad a pesar de los fallos encontrados.

Bugtraq pertenecía a SecurityFocus, que a su vez pertenece hoy al fabricante de las suites de seguridad de Symantec.

Puede darse de alta en la lista de distribución en el sitio: http://www.securityfocus.com/archive

Página de inicio de SecurityFocus

c. Los boletines de las distribuciones

Los editores de las mayores distribuciones facilitan también boletines de seguridad. Reutilizan a su vez las alertas de seguridad de otros organismos como el CERT o las listas de Bugtraq, pero, dado que cada editor es responsable de los paquetes que publica, debe publicar él mismo la correspondiente corrección y es habitual que parchee algunos productos y éstos difieran del original. De ahí la emisión de una alerta para prevenir a sus clientes y usuarios. A continuación mostramos los sitios donde puede obtener información de seguridad para las principales distribuciones:

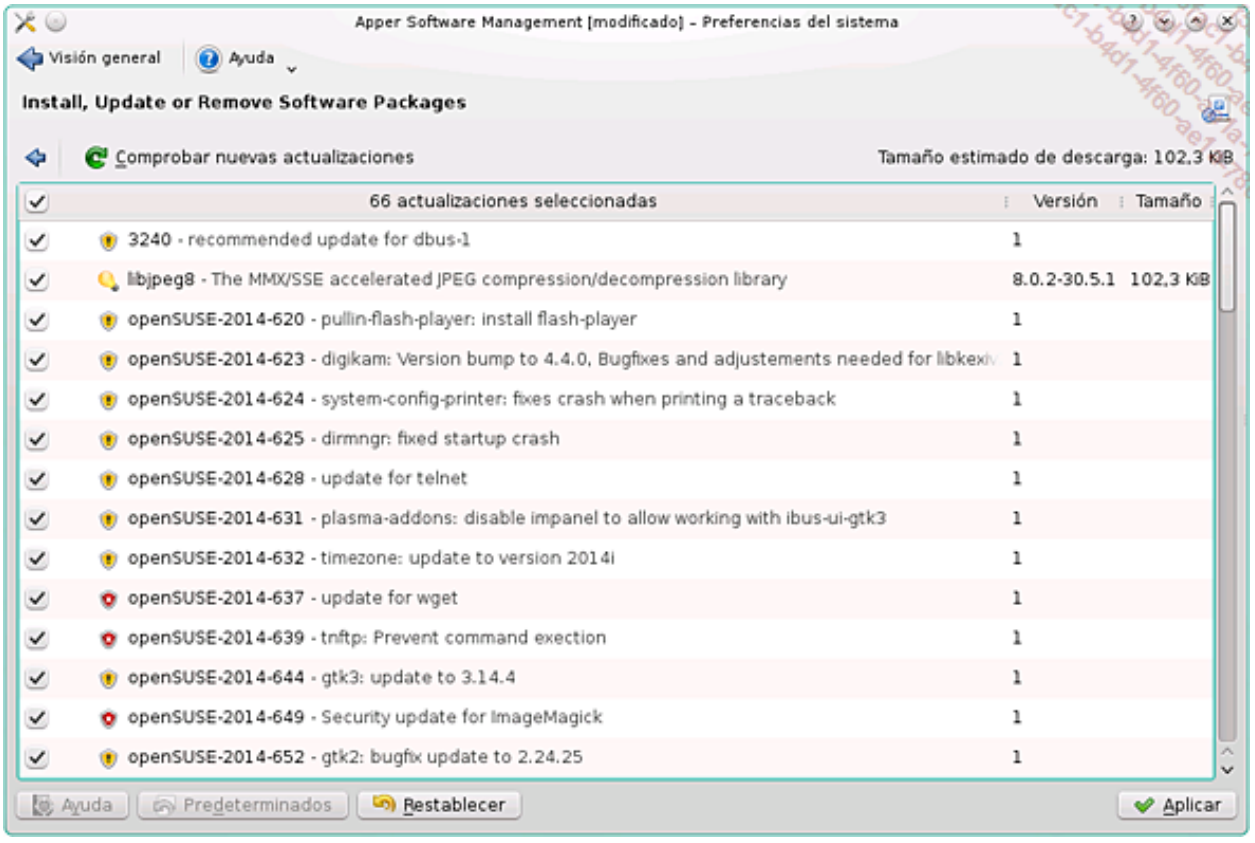
- Debian: http://www.debian.org/security/
- openSUSE: http://lists.opensuse.org/opensuse-security-announce/

- Fedora: https://www.redhat.com/archives/fedora-security-list/
- Ubuntu: http://www.ubuntu.com/usn
- Mandriva: http://www.mandriva.com/security/advisories
- Red Hat Enterprise: http://www.redhat.com/security/updates/errata/

d. Los parches correctores

No basta con dar una alerta cuando se detecta un agujero de seguridad; también hace falta arreglarlo. Para ello, los editores suministran o bien paquetes corregidos (el vínculo se encuentra a menudo en el boletín de alerta), o bien parches correctores. Las distribuciones suministran habitualmente un componente que permite recuperar esos parches e informarle sobre su disponibilidad.

Cada distribución dispone de una herramienta que permite comprobar e informar al usuario de las actualizaciones del sistema. Esta herramienta puede ser distinta según la distribución; puede consultar la documentación proporcionada por el autor de la distribución para conocerla. Sin embargo, el uso de PackageKit (http://www.freedesktop.org/software/PackageKit/) es el más común, por lo menos su módulo de actualización. Es el caso de Ubuntu, openSUSE (y los productos Novell) y Fedora. Iniciado en su interfaz gráfica, le informa desde la barra de tareas o la zona de notificación de las actualizaciones disponibles para su entorno. El siguiente ejemplo muestra la ejecución de Apper (módulo de PackageKit para KDE anteriormente llamado kPackageKit), que anuncia la existencia de varios parches de seguridad en openSUSE.



Detección de varias actualizaciones de seguridad por Apper

Seguridad de servicos y de red

1. Comprobar los puertos abiertos

a. Los sockets

Las conexiones de red entre dos máquinas se efectúan mediante **sockets.** Un socket es una conexión entre dos puntos empleando una red. Una máquina dispone de una dirección IP y de puertos (virtuales) de conexión numerados, a los cuales se vinculan servicios (ver el capítulo La red). Un cliente establece una conexión desde un puerto de su máquina (número de puerto > 1024, en general elegido de manera aleatoria entre los puertos libres) hacia un puerto determinado de otra máquina, por ejemplo un servidor web en el puerto 80. La comunicación establecida entre los dos pasa por un socket. Se puede configurar el sistema para aceptar o rechazar conexiones desde o hacia determinados puertos locales o distantes; igualmente para las direcciones IP. Es el papel del firewall (cortafuergos) como, por ejemplo, **Netfilter**.

En una instalación Linux típica no se activan los cortafuegos, excepto si dispone de esa opción durante la instalación. No se filtran los puertos y cada máquina exterior puede intentar establecer una conexión a la suya a través de un puerto: eso se denomina "abrir un socket".

No significa necesariamente que haya un agujero de seguridad: si no hay servicio alguno a la escucha, la conexión es imposible. En situación normal, los equipos inician muchos servicios por defecto. Si algunos padecen vulnerabilidades, o se han configurado de manera demasiado laxa, se produce un riesgo real de intrusión.

b. Información desde netstat

El capítulo La red le ha presentado la herramienta **netstat**, que permite obtener información y estadísticas de red sobre una máquina local. En particular se puede comprobar cuáles son los puertos a la escucha en su máquina, quién ha establecido una conexión y qué procesos locales (servicios) están a la escucha:

```
# netstat -a -A inet -p
Conexiones Internet activas (servidores y establecidas)
Proto Recv-Q Send-Q Dirección local          Dirección remota
Estado        PID/Program name
tcp        0      0 *:5800                  *:*
LISTEN      31232/kded [kdeinit
tcp        0      0 *:5801                  *:*
LISTEN      23224/xinetd
tcp        0      0 *:vnc-server            *:*
LISTEN      31232/kded [kdeinit
tcp        0      0 *:5901                  *:*
LISTEN      23224/xinetd
tcp        0      0 *:sunrpc                *:*
LISTEN      3076/portmap
tcp        0      0 *:ndmp                  *:*
LISTEN      26746/sshd
tcp        0      0 *:6000                  *:*
LISTEN      31088/Xorg
tcp        0      0 *:7634                  *:*
LISTEN      3006/hddtemp
tcp        0      0 *:ftp                    *:*
LISTEN      26772/vsftpd
tcp        0      0 localhost:ipp           *:*
LISTEN      3092/cupsd
tcp        0      0 localhost:smtp          *:*
LISTEN      3159/master
tcp        0      0 slyserver:38144         eeepc:ssh
```

```
ESTABLISHED 25622/ssh
tcp        0      0 slyserver:45046         eeepc:ms-wbt-server
ESTABLISHED 7803/rdesktop
tcp        0      0 slyserver:ftp           eeepc:49502
FIN_WAIT2   -
udp        0      0 *:xdmcp                 *:*
  27850/kdm
udp        0      0 *:59742                 *:*
  2984/avahi-daemon:
udp        0      0 *:mdns                  *:*
  2984/avahi-daemon:
udp        0      0 *:sunrpc                *:*
  3076/portmap
udp        0      0 *:ipp                   *:*
  3092/cupsd
```

Las líneas en negrita muestran tres conexiones establecidas o terminadas entre las máquinas slyserver y eeepc.

- el eeepc estaba conectado en el puerto ftp de slyserver,
- slyserver está conectado en el puerto ssh del eeepc,
- slyserver está conectado al puerto ms-wbt-server del eeepc (protocolo terminal server).

c. La herramienta nmap

Existe un batallón de herramientas de seguridad, de comprobación, de pruebas de fiabilidad, etc. La herramienta **nmap** forma parte de él. Se define como una herramienta de exploración de red y de control de seguridad. Permite probar las conexiones de red de una máquina determinada y obtener mucha información. En particular, analizando tramas, logra determinar el tipo y la versión del sistema operativo remoto.

El manual de instrucciones de nmap tiene más de 2000 líneas, por lo que es imposible estudiarlo todo. A continuación mostramos algunas de sus posibilidades:

Examine los puertos a la escucha en la máquina de prueba que se ha usado durante la redacción de este libro. Varios puertos están abiertos (unos servicios están a la escucha). Algunos de ellos pueden presentar riesgos: telnet, netbios-ssn y microsoft-ds (recursos compartidos de Windows), ftp, vnc, etc.

```
# nmap localhost

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 14:41 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 1684 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
25/tcp    open  smtp
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
5800/tcp  open  vnc-http
5801/tcp  open  vnc-http-1
5900/tcp  open  vnc
5901/tcp  open  vnc-1
6000/tcp  open  X11
10000/tcp open  snet-sensor-mgmt

Nmap finished: 1 IP address (1 host up) scanned in 0.170 seconds
```

En algunos casos, es posible desactivar los servicios innecesarios:

```
# chkconfig telnet off
# service vsftpd stop
# service smb stop
# service nmb stop
# service xinetd restart
# nmap localhost
Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 14:48 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 1688 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
111/tcp   open  rpcbind
631/tcp   open  ipp
5800/tcp  open  vnc-http
5801/tcp  open  vnc-http-1
5900/tcp  open  vnc
5901/tcp  open  vnc-1
6000/tcp  open  X11
10000/tcp open  snet-sensor-mgmt
```

En cuanto se detiene un servicio de red, ya no se puede acceder al puerto.

El parámetro –A permite detectar además el sistema operativo remoto y su versión. Para esto, uno o varios puertos deben estar abiertos. Más aún; nmap interroga a cada servicio asociado a los puertos encontrados para recuperar información. Abajo y en negrita resaltamos los valores más relevantes:

```
# nmap -A machine

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-22 19:54 CEST
Interesting ports on machine.midominio.com (192.168.1.25):
Not shown: 1676 closed ports
PORT     STATE SERVICE     VERSION
9/tcp    open  discard
13/tcp   open  daytime
21/tcp   open  ftp         vsftpd 2.0.5
22/tcp   open  ssh         OpenSSH 4.3p2 Debian 9 (protocol 2.0)
25/tcp   open  smtp        Postfix smtpd
37/tcp   open  time         (32 bits)
53/tcp   open  domain
80/tcp   open  http        Apache httpd 2.2.3 ((Debian) PHP/5.2.0-8+etch10)
111/tcp  open  rpcbind      2 (rpc #100000)
113/tcp  open  ident       OpenBSD identd
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: SLYNET)
199/tcp  open  smux        Linux SNMP multiplexer
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: SLYNET)
606/tcp  open  mountd       1-2 (rpc #100005)
631/tcp  open  ipp         CUPS 1.2
901/tcp  open  http        Samba SWAT administration server
1389/tcp open  ldap        OpenLDAP 2.2.X
2049/tcp open  nfs          2 (rpc #100003)
3128/tcp open  squid-http
7937/tcp open  nsrexec      1 (rpc #390113)
7938/tcp open  rpcbind      2 (rpc #100000)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.14 - 2.6.17
Uptime: 112.678 days (since Wed Jan 30 21:39:53 2008)
Network Distance: 2 hops
Service Info: Host:  machine.midominio.org; OSs: Unix, Linux, OpenBSD
```

```
OS and Service detection performed. Please report any incorrect results
at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 115.360 seconds
```

Esto quiere decir que una máquina dispone de muy pocos secretos si no está bien protegida. El servidor encontrado tiene como sistema operativo Linux Debian Etch, que dispone de un núcleo comprendido entre 2.6.14 y 2.6.17 (un error, porque el núcleo realmente es un 2.6.18). Los servicios con los que cuenta son:

- OpenSSH 4.3p2
- Apache 2.2.3 y PHP 5.2.0
- Postfix
- Samba 3.x
- Cups 1.2
- OpenLDAP 2.2.X
- Vsftpd 2.0.5

Basta con consultar los boletines de seguridad para ver si una de estas versiones es conocida por tener problemas de seguridad; si es así, su servidor presenta riesgos.

2. Suprimir los servicios inútiles

a. Cuestiones generales

Si alguien "hackea" su sistema, es que hay alguien malintencionado que ha encontrado la manera de entrar. En contra de la idea generalizada, la instalación de un firewall no resuelve todos los problemas: más aún cuando, en una estación de trabajo, la tendencia es abrir varios puertos de redes hacia Internet (o mejor dicho, desde Internet): ftp, http, p2p (redes eDonkey, o sea eMule, Bittorrent, etc.), ssh y así sucesivamente. Ahora bien, con que un único servicio iniciado tenga un agujero de seguridad es suficiente para que su máquina sea atacada y aumenten los problemas.

Aunque el servicio no sea muy conocido, es posible que la configuración que haya aplicado sea demasiado sencilla o laxa. No sería muy inteligente dejar que su servidor ssh aceptara las conexiones desde el exterior si su contraseña o la del root es joselito, password o algo similar. Durante un ataque, el autor de este libro tuvo la oportunidad de constatar que el atacante intentaba conectarse en bucle, mediante ssh, utilizando una serie de logins/contraseñas predefinidos entre varias combinaciones clásicas preconfiguradas por defecto en determinados programas.

Un caso sencillo es el de una instalación de MySQL por defecto donde la cuenta de administración no tiene contraseña.

Evalúe desactivar todos los servicios que no necesita. Si resulta que necesita algunos en ciertos momentos, y en otros no, no dude en iniciarlos sólo cuando los necesite para pararlos después. Asimismo, en su firewall (netfilter u otro), deje abiertos sólo los puertos estrictamente necesarios.

b. Servicios standalone

Los servicios standalone, es decir, iniciados de manera independiente, están controlados mediante el comando **service** o **`/etc/init.d/service`**. Para controlar las paradas y reinicio de estos servicios de manera permanente, utilice los comandos **chkconfig** (distribuciones RPM), **update-rc.d**(Debian), **systemctl enable/disable** (systemd).

c. Servicios xinetd

Se pueden activar y desactivar los servicios controlados por xinetd con la opción **disable** de su archivo de configuración.

```
$ pwd
/etc/xinetd.d
$ grep disable *
chargen:        disable        = yes
chargen-udp:    disable        = yes
cups-lpd:       disable        = yes
...
vmware-authd:   disable        = no
vnc:            disable        = yes
...
```

Para actualizar los cambios, fuerce **xinetd** para que vuelva a cargar su configuración.

```
# /etc/init.d/xinetd reload
```

3. Los tcp_wrappers

Las **envolturas TCP** o simplemente **tcp_wrappers** permiten la comprobación de los accesos a un servicio de red determinado (service, xinetd, portmapper). Cada programa que utiliza los**tcp_wrappers** se compila con la librería **libwrap** de manera estática (el comando ldd no permite ver la librería).

Para saber si un servicio de red está compilado con libwrap, se introduce el comando siguiente:

```
strings -f <binario> | grep hosts_access
```

Aquí tenemos un ejemplo con **xinetd**, que utiliza las **tcp_wrappers**:

```
# strings -f /usr/sbin/xinetd |grep hosts_access
/usr/sbin/xinetd: hosts_access
```

Si no se devuelve ninguna línea, el programa no utiliza las **tcp_wrappers**.

Entre los servicios que utilizan las **tcp_wrappers**, encontramos:

- **sendmail** (incluyendo postfix);
- **sshd** (ssh);
- **xinetd** (y por lo tanto de manera indirecta todos los servicios asociados);
- **vsftpd** (ftp);
- **portmap** (y por lo tanto nis, nfs);
- **in.telnetd** (telnet), así como la mayoría de los servicios soportados por xinetd;
- **dovecot** (imap, pop).

La comprobación de acceso a un servicio embebido TCP se hace en tres etapas:

- ¿se autoriza el acceso de manera explícita?
- si no es el caso, ¿se prohíbe el acceso de manera explícita?
- si no es el caso, por defecto, se autoriza el acceso.

Los archivos de configuración son **`/etc/hosts.allow`** y **`/etc/hosts.deny`**. La sintaxis es común:

```
daemon_list: client_list [:options]
```

- **daemon_list**: lista de los **ejecutables (NO DE LOS SERVICIOS)** separados por comas. Puede poner **ALL** para especificar todos los servicios. Si se dispone de varias interfaces red, se puede usar la sintaxis con @: servicio@ip.

```
in.telnetd: ...
sshd, portmap: ...
sshd@192.168.1.7: ...
```

- **client_list**: clientes autorizados o prohibidos para este servicio. Se puede especificar la dirección IP, el nombre, la máscara de red, el nombre de la red, etc.

```
... : 192.168.1.7, 192.168.1.8
... : 192.168.1.
... : puesto1, puesto2
... : 192.168.1.0/255.255.255.0
... : .midominio.org
```

La lista de clientes admite una sintaxis avanzada:

- **ALL**: correspondencia sistemática.
- **LOCAL**: todos los anfitriones cuyo nombre no contiene punto (puesto1, puesto2, etc.).
- **UNKNOWN**: anfitrión cuyo nombre no se puede resolver.
- **KNOWN**: anfitrión cuyo nombre se puede resolver.
- **PARANOID**: anfitrión cuyo nombre no se puede resolver o cuyo IP no tiene resolución inversa.
- **EXCEPT**: permite excluir ciertos anfitriones.

```
ALL EXCEPT puesto10
```

Para verificar una regla, el sistema lee primero **/etc/hosts.allow**, luego **/etc/hosts.deny**. La búsqueda se detiene en la primera correspondencia encontrada. Una línea en hosts.allow autoriza la conexión. Una línea en hosts.deny prohíbe la conexión. Si no se deniega de manera explícita el acceso, se autoriza: la petición no corresponde a ningún criterio.

En el ejemplo siguiente:

- Sólo los miembros de la subred 192.168.1.0 tienen permiso para conectarse al servidor ftp (prohibido para todos los demás).
- Los anfitriones puesto 1 y puesto 2 tienen acceso a telnet y portmap.
- Los anfitriones de baddominio.org, excepto trusted, no tienen conexión alguna posible.
- Se prohíbe el servicio pop/imap a todos los de la red 192.168.0.0, salvo 192.168.1.5.

```
# /etc/hosts.allow
vsftpd:            192.168.1.
in.telnetd, portmap:    puesto1, puesto2

# /etc/hosts.deny
ALL: .baddominio.org except trusted.baddominio.org
vsftpd,in.telnetd,portmap: ALL
dovecot : 192.168.0. EXCEPT 192.168.1.5
```

4. Netfilter

a. Presentación

Netfilter es una arquitectura de filtro de los paquetes para los núcleos de Linux 2.4 y 2.6. El filtrado se hace en el mismo núcleo en las capas 2, 3 y 4 del modelo OSI, es decir, los vínculos dados, red y transporte. Por ejemplo, es capaz de actuar a bajo nivel en las interfaces ethernet (2), en la pila IP (3) y en los protocolos de transporte como TCP (4). El filtrado no tiene estado: como Netfilter sólo inspecciona los encabezamientos de los paquetes, es muy veloz y no conlleva tiempo de espera.

Se pueden inspeccionar los contenidos de los paquetes (protocolos aplicativos) usando extensiones, pero este trabajo se delega a herramientas de usuario.

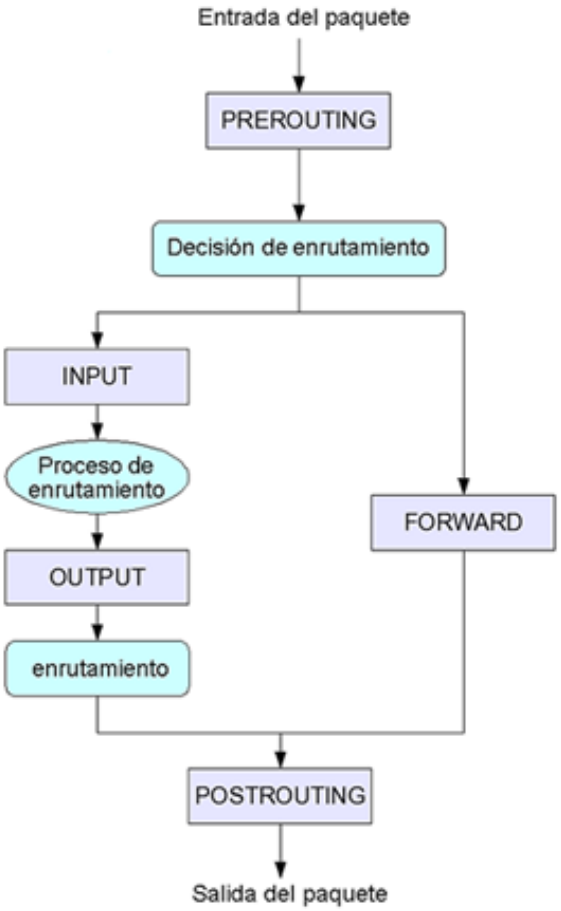
Dicho de otro modo, netfilter es un firewall que actúa a nivel del protocolo.

El programa usuario que permite actuar sobre las reglas de filtrado es **iptables**.

Se realiza la implementación a nivel del núcleo con módulos.

b. Vida de un paquete

Mejor un esquema que un largo discurso. El paquete llega por arriba y sale por abajo. Entre ambos puntos, pasa por diferentes niveles de netfilter.



Las etapas de la vida de un paquete de red con netfilter

Cada estado (rectángulo) corresponde a un punto de filtrado posible mediante el comando**iptables**.

- **PREROUTING**: trata los paquetes a su llegada. Si un paquete tiene como destino el sistema local, se le tratará con un proceso local (INPUT, OUTPUT). En caso contrario, y si está activado el forwarding, se aplicarán las reglas FORWARD y POST_ROUTING.
- **FORWARD**: los paquetes sólo cruzan el sistema local. Trata los paquetes enrutados a través del sistema local.
- **INPUT**: trata los paquetes destinados al sistema local, en entrada (después del

enrutamiento).

- **OUTPUT**: trata los paquetes que dejan el sistema local, antes POSTROUTING.
- **POSTROUTING**: trata los paquetes justo antes de su salida del sistema.

c. Principio de las reglas

Cuando netfilter trata un paquete, lo relaciona con un determinado número de reglas que determinan lo que se debe hacer con él.

- Se ordenan las reglas: la posición de una regla en una lista indica si se utilizará la regla y cuándo se hará.
- Se prueban los paquetes con cada una de las reglas, una tras otra.
- Netfilter funciona según el mecanismo de la primera correspondencia. Si una regla corresponde, se desatienden las otras y se detiene la comprobación.
- Una regla puede especificar varios criterios.
- Para que una regla corresponda a un paquete, todos los criterios deben corresponder.
- Si a pesar de todas las reglas el paquete pasa, se puede aplicar una regla por defecto.

d. Destinos de reglas

Un destino de regla determina qué acción se debe emprender cuando un paquete corresponde a los criterios de una regla. Se utiliza la opción **`-j`** de **iptables** para especificar el destino.

Los dos destinos básicos son DROP y ACCEPT. netfilter cuenta con extensiones que añaden otros destinos, como LOG o REJECT.

- **DROP**: se rechaza el paquete. No se manda ninguna notificación a la fuente.
- **REJECT**: se rechaza el paquete. Notifica un error a la fuente.
- **ACCEPT**: se acepta el paquete.
- **LOG**: se manda una información de seguimiento al syslog.

Puede crear reglas sin destino. En este caso, la regla incrementará el recuento de un contador de paquetes y un contador de bytes asociados a la regla con objetivos estadísticos.

e. Primer ejemplo

A continuación se presenta una regla sencilla que prohíbe los paquetes procedentes de 192.168.1.11.

```
# iptables -A INPUT -s 192.168.1.11 -j DROP
```

- **-A**: punto de filtrado (INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING), llamado también **cadena**.
- **-s**: fuente, puede ser una dirección IP, un nombre de anfitrión, una red, etc.
- **-j**: jump, objetivo o destino de la regla (ACCEPT, DROP, REJECT...)

Resultado: prohibirá la entrada de los paquetes cuya fuente es 192.168.1.11.

f. Operaciones básicas

Se numeran las reglas a partir de 1.

- Añada una regla con **-A**.

```
# iptables -A INPUT -s 192.168.1.2 -j DROP
```

- Inserte una regla con **-I** en la posición deseada.

```
# iptables -I OUTPUT -d 192.168.1.25 -j DROP 3 # insertar a la 3ª posición
```

- Suprima una regla con **-D**.

```
# iptables -D INPUT 1 # suprima la regla 1
```

- Liste las reglas con **-L**.

```
# iptables -L OUTPUT
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- Utilice **-F** (flush) para suprimir el conjunto de las reglas.

```
# iptables -F
```

- Utilice **-P** (policy) para modificar las reglas por defecto de una cadena.

```
# iptables -P INPUT DROP
# iptables -L INPUT
Chain INPUT (policy DROP)
target     prot opt source               destination
```

g. Criterios de correspondencia

General

Los criterios de correspondencia determinan la validez de una regla. Se deben comprobar todos los criterios para bloquear un paquete. Los criterios básicos son:

- **-i**: interfaz entrante (filtrado de capa 2);
- **-o**: interfaz saliente (filtrado de capa 2);
- **-p**: protocolo de capa 4. Para los nombres, ver el archivo **`/etc/protocols`**;
- **-s**: dirección IP de la fuente (o red);
- **-d**: dirección IP de destino (o red).

- Prohibir las entradas por eth0.

```
iptables -A INPUT -i eth0 -j DROP
```

- Prohibir el forward entre eth1 y eth2.

```
iptables -A FORWARD -i eth1 -o eth0 -j DROP
```

- Prohibir el protocolo ICMP en entrada (¡el ping!).

```
iptables -A INPUT -p icmp -j DROP
```

TCP, UDP e ICMP

Siguiendo el protocolo (capa 4), algunas opciones son posibles. Es el caso de tcp, udp o icmp (en particular utilizado por ping). Se realiza en general el filtrado a nivel de protocolo con extensiones a netfilter.

- **-p**: protocolo (tcp, udp, icmp, etc.)
- **--sport**: puerto fuente
- **--dport**: puerto destino

Si desea prohibir, por ejemplo, las conexiones entrantes con destino al puerto 80 (servidor httpd), proceda como a continuación:

```
iptables -A INPUT -p tcp --dport 80 -j DROP
```

Argumentos de los criterios

Para las direcciones, puede especificar:

- un anfitrión por su nombre o su dirección IP;
- una red por su nombre o su máscara (192.168.1.0/24, 192.168.1.0/255.255.255.0).

Para los puertos:

- un número;
- un nombre (ver/etc/servicios);
- una gama de puertos: **123:1024**.

En todos estos casos puede usarse el signo de exclamación para definir excepciones.

Para prohibir en la entrada todas las conexiones salvo las de 10.0.0.1:

```
# iptables -A INPUT -s ! 10.0.0.1 -j DROP
```

Usuarios y grupos

Netfilter gestiona los UID y GID de los procesos que emiten paquetes desde el host. Puede realizar un filtrado en la salida para los usuarios y grupos locales, para las reglas OUTPUT y POSTROUTING. El módulo es **owner**. Los usuarios se especifican con **--uid-owner** y los grupos con **-gid-owner**.

Esto puede ser muy útil, por ejemplo, en el caso de un proxy transparente. Se puede redirigir cualquier conexión saliente con destino el puerto 80 hacia el puerto 3128 para todo usuario excepto el proxy. Para ello hay que ejecutar el siguiente comando:

```
# iptables -t nat -A OUTPUT -m owner ! --uid-owner proxy -p tcp
-m tcp --dport 80 -j REDIRECT --to-ports 3128
```

Para permitir a un usuario pasar por otra regla, puede ejecutar el siguiente comando ANTES (recuerde que el orden es importante) de la redirección:

```
# iptables -t nat -A OUTPUT -p tcp --dport 80 -m owner --uid-owner seb -j
ACCEPT
```

h. Tablas

Además de los puntos de filtrado (INPUT, OUTPUT, etc.) Netfilter define tablas particulares que pueden contener reglas específicas. Puede crear sus propias tablas que contengan sus propias reglas y a continuación, adjuntarlas a un punto de filtrado. Pero Netfilter también dispone de tablas

por defecto y especialmente la tabla **nat**, que contiene las reglas que permiten modificar las IP y puertos origen y destino de los paquetes. Se muestra del siguiente modo:

```
# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:www owner UID match root
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:www owner UID match seb
ACCEPT     tcp  --  anywhere             anywhere
tcp dpt:www owner UID match esteban
REDIRECT   tcp  --  anywhere             anywhere
! owner UID match proxy tcp dpt:www redir ports 3128
...
```

La siguiente regla modifica dinámicamente la IP de destino de un cierto tipo de paquetes. Los paquetes con destino el puerto 80 de la IP 192.168.1.2, se modificarán de tal modo que se reenviarán a la 192.168.10.10:

```
# iptables -t nat -A OUTPUT --dst 192.168.1.2 -p tcp --dport 80 -j
DNAT --to-destination 192.168.10.10
```

i. Guardar las configuraciones

Las reglas definidas con **iptables** se cargan en memoria y se transmiten de manera dinámica al núcleo. Pero al arrancar de nuevo la máquina, se pierden. **Red Hat** permite guardar el conjunto de las reglas de manera que se vuelvan persistentes.

```
# service iptables save
```

Las reglas están guardadas en el archivo **/etc/sysconfig/iptables**.

En Debian y Ubuntu, se pueden utilizar las herramientas **iptables-save** e **iptables-restore**, guardando las reglas por ejemplo en **/etc/iptables.rules**. Puede entonces emplear las líneas siguientes en **/etc/network/interfaces**:

```
pre-up iptables-restore < /etc/iptables.rules
post-down iptables-restore < /etc/iptables.downrules
```

Para las demás distribuciones, consulte la documentación. En algunos casos, puede ser útil crear su propio script.

Se deberían cargar las reglas **iptables ANTES** de la activación de la red. Así, no hay ningún riesgo de seguridad, porque las reglas serán directamente válidas en el momento en que se active la red.

5. UFW

Partiendo de la premisa que iptables es un poco complicado para el usuario medio o incluso un poco avanzado, los desarrolladores de la distribución Ubuntu han decidido crear una herramienta en línea de comandos que permita configurar de forma más sencilla Netfilter. UFW (*Uncomplicated Firewall*) es una capa frontal de Netfilter e iptables. No los reemplaza, sino que permite crear en su lugar reglas Netfilter. Utiliza y es un complemento de iptables. Sólo hay que usar un comando: **ufw**.

UFW está disponible para Ubuntu y también puede instalarse en Debian. Si no está disponible por defecto en otras distribuciones, se puede recompilar, las fuentes están disponibles. Se están dedicando esfuerzos para crear paquetes, especialmente para Fedora. Los comandos siguientes se deben ejecutar directamente como root o desde sudo.

a. Activación y estado

Active ufw con el parámetro **enable**, como se muestra a continuación:

```
# ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
```

En Ubuntu, ufw se inicia mediante init. El script de init comprobará el estado de activación de ufw para determinar si debe cargar las reglas o no hacerlo. De este modo, el script init siempre debe estar activo, pero es esta llamada al comando ufw la que determina si debe funcionar o no.

El estado actual se obtiene con el parámetro **status**. En el ejemplo siguiente ya hay una regla presente (la de acceso a través del puerto ssh):

```
# ufw status
Estado: activo

Hasta                      Acción      Desde
-----                      -------     -------
22                         ALLOW       Anywhere
```

Se puede añadir el parámetro **verbose** para obtener más detalles, especialmente para mostrar el estado por defecto de las distintas tablas:

```
# ufw status verbose
Estado: activo
Acceso: on (low)
Por defecto: deny (Entrada), allow (Salida)
```

ufw se desactiva con disable. Las reglas creadas con ufw se descargarán. Sin embargo, las reglas creadas por iptables no se modificarán.

```
# ufw disable
El cortafuegos está detenido y desactivado en el arranque del sistema
```

b. Reglas por defecto

Con ufw, siendo voluntariamente sencillo, sólo el tráfico entrante y saliente tiene reglas por defecto: allow o deny para los permisos, como en los ejemplos siguientes:

```
# ufw default deny incoming
La política incoming predeterminada cambió a «deny»
(asegúrese de actualizar sus reglas consecuentemente)
# ufw default allow outgoing
La política outgoing predeterminada cambió a «allow»
(asegúrese de actualizar sus reglas consecuentemente)
```

El registro se activa y se desactiva así:

```
# ufw logging off
Registro desactivado
# ufw logging on
Registro activado
```

c. Gestión de reglas

Reglas simples

La forma más sencilla consiste en permitir o prohibir conexiones entrantes o salientes en función de los puertos. Por ejemplo, puede permitir el acceso al servicio ssh de su servidor:

```
# ufw allow 22
Regla añadida
```

Mejor que el número del puerto, puede utilizar el nombre del servicio tal y como se define en /etc/services.

Para prohibir el acceso con deny:

```
# ufw deny 22
Regla actualizada
```

Eliminación

Hay veces en que no nos es suficiente, ya que este comando no elimina la regla antigua, la actualiza. Para eliminar definitivamente una regla, muéstrelas todas de forma numérica, como se muestra a continuación:

```
# ufw satus numbered
Estado: activo

     Hasta                      Acción      Desde
     -----                      ------------------------
[1] Anywhere                    ALLOW IN    212.27.38.253/udp
[2] 22                          DENY IN     Anywhere
```

Elimine la regla 2:

```
# ufw delete 2
Borrando:
 deny 22
¿Continuar con la operación (s|n)? s
Regla eliminada
```

También puede borrar la regla explícitamente:

```
# ufw delete deny 22
Regla eliminada
```

Aplicaciones

Debido a que ufw tiene que ser sencillo a ojos del usuario, las aplicaciones instaladas pueden ofrecer reglas predefinidas para simplificar la integración de reglas asociadas. Por ejemplo, el uso de un servidor Samba requiere la apertura de algunos puertos. En la instalación del paquete asociado, los scripts de postinstalación llamarán a ufw para definir un perfil que podrá activar a continuación. La lista de aplicaciones para las que hay reglas se muestra con **app list**:

```
# ufw app list
Aplicaciones disponibles:
  CUPS
  OpenSSH
```

```
  Postfix
  Postfix Submission
  Samba
  Squid
```

Los detalles de una aplicación se muestran con app info. Por ejemplo, consultamos los detalles de Samba:

```
# ufw app info Samba
Perfil: Samba
Título: LanManager-like file and printer server for Unix
Descripción: The Samba software suite is a collection of programs that
implements the SMB/CIFS protocol for unix systems, allowing you to serve
files and printers to Windows, NT, OS/2 and DOS clients. This protocol is
sometimes also referred to as the LanManager or NetBIOS protocol.

Puertos:
  137,138/udp
  139,445/tcp
```

Las reglas se activan o desactivan fácilmente, tal y como se muestra a continuación:

```
# ufw allow Samba
Regla añadida
# ufw deny Samba
Regla actualizada
```

Reglas más complejas

Para cada puerto, puede especificar el protocolo:

```
# ufw allow 22/tcp
```

También puede utilizar una sintaxis más extendida, como la que prohíbe todo el tráfico entrante en TCP desde la subred 192.168.1.0/24 hacia cualquier IP de su servidor el puerto 22:

```
# ufw deny proto tcp from 192.168.1.0/24 to any port 22
```

La regla siguiente permitirá el acceso a los puertos 80, 443 y del 8080 al 8090 en su ordenador:

```
# ufw allow proto tcp from any to any port 80,443,8080:8090
```

Para prohibir cualquier conexión desde su equipo hacia cualquier equipo al puerto 80:

```
# ufw reject out to any port 80
```

d. Limitaciones

UFW es práctico para casos relativamente simples: servidores básicos, puestos de trabajo, ordenadores personales. Desafortunadamente, para un uso avanzado, UFW es insuficiente. Por ejemplo, los protocolos gestionados están limitados a TCP y UDP. Si desea autorizar multicast para protocolos como VRRP u OSPF, siga buscando. En ese caso deberá utilizar netfilter y los comandos vistos previamente.

6. GPG

a. Un clon de PGP

GPG (*Gnu Privacy Guard*) es un clon libre de **PGP** (*Pretty Good Privacy*). Implementa el algoritmo de cifrado RSA. PGP es objeto de una norma que GPG respeta. Esto significa que las dos implementaciones son compatibles: las claves generadas por uno u otro se pueden intercambiar.

El objetivo de GPG es cifrar una comunicación gracias a un cifrado por claves asimétricas. Una clave permite firmar el texto, otra clave sirve para encriptar el texto.

El cifrado por claves asimétricas utiliza dos claves, una pública y otra privada:

- Su **clave pública**, difundida públicamente, permite al destinatario descifrar el mensaje que ha firmado (con la clave privada).
- Su **clave privada** le permite firmar un mensaje. La persona que lo recibe comprueba su firma con la ayuda de la clave pública que le ha proporcionado, demonstrando así que usted es realmente el autor del mensaje.

A continuación, vamos a presentar todo lo necesario para crear y gestionar un almacén de claves: sus claves públicas y privadas, las claves públicas de sus amigos, las firmas, etc.

Se presentan los comandos en modo consola. Existen herramientas gráficas, como **kgpg**, para gestionar las claves. Asimismo, numerosos clientes de mensajería integran GPG (Kmail lo integra por defecto). Gracias al plugin Enigmail se puede incorporar facilmente GPG a Thunderbird. Se utilizan también las firmas para comprobar el origen de los paquetes RPM, por ejemplo.

b. Generar claves

Utilice gpg con el parámetro `--gen-key`. GPG empieza creando, si no existen ya, el directorio raíz de GPG propio de cada usuario y los archivos que contendrán los elementos gestionados por gpg.

```
seb@slyserver:~> gpg --gen-key
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directorio '/home/seb/.gnupg' creado
gpg:  nuevo archivo de configuración '/home/seb/.gnupg/gpg.conf' creado
gpg: ADVERTENCIA: las opciones de '/home/seb/.gnupg/gpg.conf' aún no están
activas esta vez
gpg: el llavero '/home/seb/.gnupg/secring.gpg' ha sido creado
gpg: el llavero '/home/seb/.gnupg/pubring.gpg' ha sido creado
```

Seleccione el tipo de clave deseada. La primera elección incluye las otras dos; por lo tanto, es la que debería efectuarse:

```
Seleccione el tipo de clave deseado:
   (1) DSA y Elgamal (por defecto)
   (2) DSA (firma sola)
   (5) RSA (firma sola)
¿Su elección? 1
```

Luego debe elegir el tamaño, en bits, de la clave que hay que generar. Cuánto más larga sea la clave, más complejo es el cifrado. Pero en máquinas más antiguas, el descifrado puede tardar entonces más tiempo. Puede elegir una clave de 1024 bits por ejemplo:

```
El par de claves DSA dará 1024 bits.
las claves ELG pueden tener entre 1024 y 4096 bits de longitud.
¿Qué tamaño de clave desea? (2048) 1024
El tamaño pedido es 1024 bits
```

Luego indique la duración de validez de las claves. Por defecto, si prevé utilizarla el mayor tiempo posible, elija cero para una duración infinita. Si no, precise la duración deseada según el formato

indicado. Valide su elección. Debe saber que es posible invalidar una clave generada por error con una clave de revocación de la que hablaremos un poco más adelante.

```
Especifique cuánto tiempo debería ser válida esta clave.
           0 = la clave no expira
        <n>  = la clave expira en n días
        <n>w = la clave expira en n semanas
        <n>m = la clave expira en n meses
        <n>y = la clave expira en n años
¿La clave es válida para? (0)
La clave no expira.
¿Es correcto? (s/N) s
```

Para generar la clave, GPG le pide introducir su nombre, su dirección de correco electrónico (e-mail) y un comentario. En el nombre, no dude en poner sus apellidos y su nombre. Se podrá utilizar luego el comentario como alias para determinados comandos GPG. Estos elementos no se podrán cambiar más adelante. Por el contrario, le será posible añadir direcciones de email con el fin de asociar varias de ellas a una sola clave.

```
Necesita un nombre de usuario para identificar su clave,
el programa lo construye a partir del nombre real, de un comentario y
de una dirección de email, de la manera siguiente:
    «Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>»

Nombre real: Sébastien ROHAUT
Dirección de e-mail: sebastien.rohaut@dominio.es
Comentario: slyce
Usted utiliza el juego de caracteres 'utf-8'.
Ha seleccionado este nombre de usuario:
   "Sébastien ROHAUT (slyce) <sebastien.rohaut@dominio.es>"

Cambiar el (N)ombre, el (C)omentario, el (E)-mail o (O)K/(S)alir ? O
```

La **passphrase** o frase de contraseña complementa la clave privada. Se pedirá esta frase a cada envío o recepción de mensaje privado encriptado. También asegura, al menos mientras no la encuentren, una cierta seguridad de la clave privada, incluso si la roban, ya que esta clave no se puede utilizar sin esta frase. Elija una frase (o palabra) bastante compleja, que recuerde siempre, pero no muy larga, ya que tendrá que teclearla de manera regular.

```
Necesita una frase de contraseña para proteger su clave secreta.
Inserte la frase de contraseña:
Repita la frase de contraseña:
```

GPG efectúa la última etapa, la generación de las propias claves. GPG va a utilizar información introducida y genera números aleatorios. Le damos un consejo: durante el cálculo, haga trabajar su ordenador para que el generador de números aleatorios "espabile".

```
Se debe generar un gran número de bytes aleatorios. Tendría que
hacer otra cosa (teclear, mover el ratón, utilizar
discos) durante la generación de los primeros números. Eso da
al generador de nombres aleatorios más oportunidad de tener
bastante entropía.
gpg: clave 13E021A8 marcada como si tuviera una confianza última.
se ha creado y firmado las claves pública y secreta.

gpg: comprobar la base de confianza
gpg: 3 marginale(s) necesarias, 1 completa(s) necesarias, modelo
de confianza PGP
gpg: profundidad: 0  válida:   1  firmado:   0
confianza: 0-. 0g. 0n. 0m. 0f. 1u
pub   1024D/13E021A8 2009-05-04
```

```
Huella de la clave = 6115 DE46 2678 40AB 0AB2  B8B9 1F12 D427 13E0 21A8
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@dominio.es>
sub   1024g/BA311C5C 2009-05-04
```

Las últimas cuatro líneas resumen toda la información relativa al trabajo efectuado por GPG:

- **pub**: da la longitud de la clave (1024 bits), de número 13E021A8, generada el 04/05/2009.
- **huella**: es el fingerprint que permite determinar la validez de la clave pública, un poco como la suma md5 de cualquier archivo. Es casi imposible (pero no del todo), que claves de gran tamaño tengan una huella idéntica.
- **uid**: información nominativa (nombre, comentario o alias/seudo, dirección e-mail).
- **sub**: es el tamaño (1024 bits), el número y la fecha de generación de la clave privada.

c. Generar una clave de revocación

Generar una clave de revocación le permitirá revocar su clave, o sea, cancelar su validez. Está firmada por su clave privada. Piense en dos casos: si le han robado su clave privada, hay que cancelarla para que deje de ser válida. Si ha perdido su passphrase, también tendrá que revocar su clave. En ambos casos, tendrá que volver a generar una clave. La clave de revocación permite a todas las personas que disponen de una clave pública comprobar que la cancelan. Una vez generada la clave de revocación, consérvela en un sitio seguro, no la publique ni tampoco la pierda.

Por lo tanto, es muy importante crear una clave de revocación justo después de haber creado el par de claves pública/privada y guardarla en un lugar seguro. Eso es así por varias razones: si pierde por ejemplo su clave privada, no tendrá ya la oportunidad de generar la clave de revocación. Si se olvida de su passphrase, no podrá utilizar más su clave privada; por lo tanto, no podrá crear la clave de revocación.

En contraposición, tenga cuidado en no divulgar su clave de revocación: si se conociera, cualquiera podría revocar una clave que usted utiliza.

Inicie el comando **gpg** con `--gen-revoke` y el nombre o el comentario (alias, seudo) asociado a la clave. Teclee la razón por la cual quiere generar una clave de revocación. Dé la razón que quiera y el comentario asociado con la elección y confirme.

```
seb@slyserver:~> gpg --gen-revoke Slyce

sec  1024D/13E021A8 2009-05-04 Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>

¿Generar un certificado de revocación para esta clave? (s/N) s
elija la causa de la revocación:
  0 = Ninguna razón especificada
  1 = Se comprometió la clave
  2 = Se sustituyó la clave
  3 = No se utiliza más la clave
  Q = Cancelar
(Aquí seguramente tendría que seleccionar 1)
¿Su decisión? 1
```

Introduzca una descripción opcional; termínela con una línea vacía:

```
> Ejemplo público para libro
>
Causa de revocación: Se comprometió la clave.
Ejemplo público para libro.
¿Está de acuerdo? (s/N) s
```

Luego, tiene que teclear, como cada vez que se debe utilizar la clave privada, su frase de contraseña:

```
Necesita una frase de contraseña para desbloquear su clave
secreta para el usuario. «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
clave de 1024 bits DSA, ID 13E021A8, creada el 2009-05-04
```

Ya se ha generado la clave. El sistema le muestra las instrucciones de uso: proteja su clave.

```
salida con formato ASCII forzado.
Certificado de revocación creado.
Muévalo a un soporte que pueda guardar; si Mallory tiene
acceso a este certificado, lo puede utilizar para hacer inutilizable su clave.
Una buena idea consiste en imprimir este certificado y luego guardarlo
en otro sitio, en caso de que el soporte se vuelva ilegible. Pero cuidado:
¡el sistema de impresión de su máquina podría almacenar estos datos
y hacerlos accesibles a otras personas!
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)
Comment: A revocation certificate should follow

iGIEIBECACIFAkoAQC8bHQJFeGVtcGxlIHB1YmxpYyBwb3VyIGxpdnJlAAoJEB8S
...
jlkHZg===9xGd
-----END PGP PUBLIC KEY BLOCK-----
```

Deje su clave GPG en un soporte seguro: imprímala, póngala en un cd grabado (incluso de forma exclusiva), en un pendrive (no confíe en este soporte), en una caja fuerte o aprendrásela de memoria (si es buena).

d. Gestionar el almacén de claves

Haga la lista de las claves generadas por GPG con el parámetro `--list-key`. Para cada clave, puede ver la información relativa a las claves públicas y privadas, su propietario y la posible fecha de expiración.

```
seb@slyserver:~> gpg --list-key
/home/seb/.gnupg/pubring.gpg
----------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@dominio.es>
sub   1024g/BA311C5C 2009-05-04
...
```

Se muestran las firmas, o más bien la lista de las firmas de cada una de sus claves, por clave pública y clave privada, con la `--list-sigs` que incluye la `--list-key`. Más adelante se explicarán las firmas.

```
seb@slyserver:~> gpg --list-sigs
/home/seb/.gnupg/pubring.gpg
----------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@dominio.es>
sig 3        13E021A8 2009-05-04  Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>
sub   1024g/BA311C5C 2009-05-04
sig          13E021A8 2009-05-04  Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>
```

e. Exportar la clave pública

Para que los demás puedan utilizar su clave pública para mandar mensajes, les debe proporcionar la clave y, por lo tanto, exportarla en un formato legible. Eso se hace con `gpg --export --armor`:

```
seb@slyserver:/usr/share/man> gpg --export --armor
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)

mQGiBEn/SbsRBADYR9n4VE2k+z7Ec9uT71+2KKlFgEgajQoYUSchyqZnIov74sCe
TSDyNMFI6SAJ/FrsM1SMJgtYUMedkfjl1hTl4i2T+YzIatJC00zAuwvhzyD23xpr
Q0dqtzBlb0pUbFNOgx5gF2McbiwknYcup5dbCVLjr6kqSC8iTACseZuIPwCg1Rfd
HcsKri8zWmeAp6u1nEpu7xED/j5gFeLhBhceac42O/Wy3dNrq6omro47RL7Jldp2
...
rr0RMO6vMWxo0oqpNaWAA8igXzUKexlOGfe2VmBmiEkEGBECAAkFAkn/SbsCGwwA
CgkQHxLUJxPgIajcwwCeJKyF88yfjGQuVdtelnE3SYIs7/EAoMpfNEkgmPQyAX+C
tU/c3EHZLlxk
=4v68
-----END PGP PUBLIC KEY BLOCK-----
```

La salida está truncada porque es demasiado larga. Basta con copiar este bloque completo y mandarlo, con el método que desee, a la persona a la cual quiere facilitársela (archivo adjunto de mail, servidores de claves, publicada en un sitio, etc.).

Puede exportar su clave pública hacia un servidor de claves GPG, por defecto keys.gnupg.net. Así, una persona que desea obtener su clave pública, para firmarla por ejemplo como veremos más adelante, podrá recuperarla en el servidor correspondiente.

```
seb@slyserver:~>
gpg -k /home/seb/.gnupg/pubring.gpg
------------------------------
pub   1024D/13E021A8 2009-05-04
uid                  Sébastien ROHAUT (Slyce) <sebastien.rohaut@dominio.es>
sub   1024g/BA311C5C 2009-05-04

seb@slyserver:/usr/share/man> gpg --send-keys 13E021A8
gpg: envío de la clave 13E021A8 al servidor hkp keys.gnupg.net
```

Si luego se dirige al servirdor http://keys.gnupg.net/ y teclea **Slyce** (por ejemplo, el e-mail o el identificador de clave funcionan también), obtendrá la lista de las claves correspondientes y, entre ellas, la anterior.

f. Importar una clave

Puede importar una clave en su almacén con el parámetro `--import`. El segundo parámetro puede ser un archivo que contiene la clave. Si no lo tiene, efectúe un copiar-pegar de la clave en la consola, con la última línea que debe ser `-----END PGP PUBLIC KEY BLOCK-----`; luego teclee [Ctrl] **D**.

```
seb@slyserver:~> gpg --import
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.0

mQGiBEn/SbsRBADYR9n4VE2k+z7Ec9uT71+2KKlFgEgajQoYUSchyqZnIov74sCeTSDy
NMFI6SAJ/FrsM1SMJgtYUMedkfjl1hTl4i2T+YzIatJC00zAuwvhzyD23xprQ0dqtzBlb0pUbFNO
...
ZykVTkKarr0RMO6vMWxo0oqpNaWAA8igXzUKexlOGfe2VmBmiEkEGECAAkFAkn/SbsCGwwACgkQHxL
UJxPgIajcwwCe-JKyF88yfjGQuVdtelnE3SYIs7/EAoMpfNEkgmPQyAX+CtU/c3EHZLlxk=4v68
-----END PGP PUBLIC KEY BLOCK-----
gpg: clave 13E021A8: «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>» no ha cambiado
gpg:        Cantidad total tratada:  1
```

```
gpg:                    no cambiada: 1
```

También puede recibir una clave de un servidor:

```
seb@slyserver:~> gpg --recv-keys D0FE7AFB
gpg: petición de la clave D0FE7AFB del servidor hkp keys.gnupg.net
gpg: clave D0FE7AFB: clave pública «Josh Triplett
<josh@joshtriplett.org>» importada
gpg: 3 marginale(s) necesarias, 1 completa(s) necesarias, modelo
de confianza PGP
gpg: profundidad: 0  válida:   1  firmado:   0
confianza: 0-. 0g. 0n. 0m. 0f. 1u
gpg:        Cantidad total tratada: 1
gpg:                    importada: 1
```

Puede suprimir una clave del almacén de la manera siguiente:

```
seb@slyserver:~> gpg --delete-keys D0FE7AFB
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub  1024D/D0FE7AFB 2004-06-24 Josh Triplett <josh@joshtriplett.org>

¿Quitar esta clave del almacén? (s/N) s
```

Para terminar, puede volver a generar la fingerprint de una clave, o sea, su huella. Como vimos durante la generación de las claves, sólo tendrá la prueba de que una clave pública pertenece realmente a una persona determinada si su huella es idéntica a la que dicha persona le ha suministrado.

```
seb@slyserver:~> gpg --fingerprint slyce@slyunix.org
pub   1024D/5D022685 2009-05-06
    Huella de la clave = 57BF 3ABF F392 1616 068B  7C73 779D 6BCF 5D02 2685
uid                  Sebastien ROHAUT (Slyce) <slyce@slyunix.org>
sub   2048g/4CA7E7CD 2009-05-06
```

g. Firmar una clave

Firmar una clave pública es certificar que esta clave pertenece realmente a la persona indicada por el identificador. Por eso hay que comprobar la huella antes de firmar una clave. Para firmar la clave que le ha transmitido, por ejemplo la de su amigo Esteban, tiene que importarla primero a su propio almacén:

```
seb@slyserver:~> gpg --import
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.9 (GNU/Linux)

mQGiBEoB63QRBACrfwOSo2EKJRYF0lh0Y06X2ZNNEq8+50IJmXYTXLx4b+niYzjf
V1pW5/0Of1/iM80RFU8NjkIKuxKnK0jji/WQbwAyWcTVMQ8CNrH2Fgbahi+8mn/Z
...
dAIbDAUJAA0vAAAKCRD1IZKWFJB27BVSAJoCQ1i49aTNi3NopE+zgiZYwYSXaACe
LFxBG3QJUkmLpSX8L/9vzjAZ+6Y==QMLk
-----END PGP PUBLIC KEY BLOCK-----
gpg: clave 149076EC: clave pública «Esteban (Clave de prueba)
<esteban@tele2.es>» importada
gpg:        Cantidad total tratada: 1
gpg:                    importada: 1
```

Compruebe que todo se ha desarrollado correctamente haciendo la lista de las claves:

```
seb@slyserver:~> gpg -k Esteban
pub   1024D/149076EC 2009-05-06 [expira: 2009-05-16]
uid                  Esteban (Clave de prueba) <esteban@tele2.es>
sub   2048g/ABC8FBE2 2009-05-06 [expira: 2009-05-16]
```

Recupere la huella de la clave de Esteban para compararla con la que le ha facilitado:

```
seb@slyserver:~> gpg --fingerprint Esteban
pub   1024D/149076EC 2009-05-06 [expira: 2009-05-16]
    Huella de la clave = B50E 5C76 C0A6 4BDC F83E  FB8F F521 9296 1490 76EC
uid                  Esteban (Clave de prueba) <esteban@tele2.es>
sub   2048g/ABC8FBE2 2009-05-06 [expira: 2009-05-16]
```

Si la huella es correcta, edítela: eso abre un pequeño intérprete de comandos para trabajar en el almacén o la clave correspondiente. Observe el estado de confianza y validez: está en "desconocido" porque la clave no está firmada:

```
seb@slyserver:~> gpg --edit-key Esteban
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


pub  1024D/149076EC  creado: 2009-05-06  expira: 2009-05-16  utilización: SC
                     confianza: desconocida         validez: desconocida
sub  2048g/ABC8FBE2  creado: 2009-05-06  expira: 2009-05-16  utilización: E
[ desconocida] (1). Esteban (Clave de prueba) <esteban@tele2.es>

Comando>
```

Teclee **trust** para dar su nivel de confianza:

```
Comando> trust
pub  1024D/149076EC  creado: 2009-05-06  expira: 2009-05-16  utilización: SC
                     confianza: desconocida         validez: desconocida
sub  2048g/ABC8FBE2  creado: 2009-05-06  expira: 2009-05-16  utilización: E
[ desconocida] (1). Esteban (Clave de prueba) <esteban@tele2.es>

Decida ahora hasta qué punto confía en este
usuario para que compruebe las claves de los demás usuarios (puede
usted comprobar su pasaporte, las huellas de varias
fuentes diferentes, etc.)

  1 = no sé o no diré
  2 = NO confío
  3 = confío un poco
  4 = confío totalmente
  5 = doy una confianza última
  m = volver al menú principal

¿Su decisión? 4

pub  1024D/149076EC  creado: 2009-05-06  expira: 2009-05-16  utilización: SC
                     confianza: entera      validez: desconocida
sub  2048g/ABC8FBE2  creado: 2009-05-06  expira: 2009-05-16  utilización: E
[ desconocida] (1). Esteban (Clave de prueba) <esteban@tele2.es>
Observe que la validez mostrada para la clave no es necesariamente
correcta hasta que usted no vuelva a arrancar el programa.
```

Observe los niveles de confianza: de 1 a 5, por orden ascendente. La traducción española es un poco sorprendente. La opción 3, por ejemplo, corresponde a "confío un poco". Si elige 5, se le pedirá una confirmación.

Teclee ahora **sign** para firmar la clave con su clave privada. Tendrá que introducir su passphrase:

```
Comando> sign

pub  1024D/149076EC  creado: 2009-05-06  expira: 2009-05-16  utilización: SC
                     confianza: entera      validez: desconocida
 Huella de la clave principal:
B50E 5C76 C0A6 4BDC F83E  FB8F F521 9296 1490 76EC

     Esteban (Clave de prueba) <esteban@tele2.es>

Esta clave expira el 2009-05-16.
¿Está seguro de que quiere firmar esta clave
con la suya? «Sébastien ROHAUT (Slyce) <sebastien.rohaut@dominio.es>»
(13E021A8)

¿Firmar realmente? (s/N) s

Necesita una frase de contraseña para desbloquear la
clave secreta para el usuario: «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
clave de 1024 bits DSA, ID 13E021A8, creada el 2009-05-04
```

Ahora la clave de Esteban está firmada.

h. Firmar y cifrar

Ahora que dispone de la clave pública de sus amigos, puede firmar un mensaje (de su parte) a su atención.

```
seb@slyserver:~> gpg --clearsign -u sebastien.rohaut@fdominio.es -a

Necesita una frase de contraseña para desbloquear la
clave secreta para el usuario: «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
clave de 1024 bits DSA, ID 13E021A8, creada el 2009-05-04

Hola, amigos
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hola, amigos
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.9 (GNU/Linux)

iEYEARECAAYFAkoB8f8ACgkQHxLUJxPgIagxlACgoB0NKICcEX5D13CCWnYz3tXt
S4oAoKxFR0AiKFb4krF/XScBWfg4Xcmo=zmms
-----END PGP SIGNATURE-----
```

Mande su mensaje en forma de texto, con la firma, al destinatario. Éste, al recibir el mensaje, hará la operación inversa: comprobará si la firma corresponde realmente a usted y al mensaje (suponiendo que el mensaje haya sido colocado en un archivo mensaje.asc). Es ese caso: gpg confirma que Sébastien ROHAUT ha firmado el mensaje.

```
esteban@slyserver:~> gpg --verify message.asc
gpg: Firma
hecha el miérc. 06 de mayo de 2009 22:24:31 CEST con la clave DSA ID 13E021A8
gpg: Firma correcta de «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
```

La etapa siguiente consiste en cifrar (encriptar) el mensaje completo, de tal manera que sólo el

destinatario pueda descifrarlo. Para que Sébastien pueda mandar un mensaje cifrado a Esteban del cual tiene la clave pública, haga lo siguiente:

```
seb@slyserver:~> gpg -r Esteban -e --armor
Hola Esteban, estoy probando el envío de un mensaje encriptado
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.9 (GNU/Linux)

hQINAzFP80qryPviEAf/eZpDHfd4KeSQ2ENECdlrgRAsDIULeDeI/ZPPbdiHJ1Wv
...
iSmi8LMJ1MixdV0PuGp/yA===0caY
-----END PGP MESSAGE-----
```

Mande el mensaje encriptado (del BEGIN al END) a Esteban. Éste, por su parte, intentará descifrarlo: debe ser el único en poder hacerlo con su clave privada.

```
esteban@slyserver:~> gpg --decrypt message.asc
Necesita una frase de contraseña para desbloquear su clave
secreta para el usuario. «Esteban (Clave de prueba) <esteban@tele2.es>»
clave de 2048 bits ELG, ID ABC8FBE2, creada el 2009-05-06 (ID clave
principal 149076EC)

gpg: cifrado con una clave de 2048 bits ELG, ID ABC8FBE2, creada el 2009-05-06
      «Esteban (Clave de prueba) <esteban@tele2.es>»
Hola Esteban, estoy probando el envío de un mensaje encriptado.
```

El mensaje que se muestra no contiene errores.

Última etapa: cifrar y firmar al mismo tiempo. Si su mensaje se encuentra en un archivo `test.msg`, teclee lo siguiente:

```
seb@slyserver:~> gpg -u sebastien.rohaut@tele2.es -r Esteban -a -e -s
test.msg

Necesita una frase de contraseña para desbloquear la clave
secreta para el usuario: «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
clave de 1024 bits DSA, ID 13E021A8, creada el 2009-05-04
```

Se ha creado un archivo test.msg.asc que contiene el mensaje cifrado y encriptado.

```
seb@slyserver:~> cat test.msg.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.9 (GNU/Linux)

hQIOAzFP80qryPviEAf/dxwdIBhRYhx/j0nxddcb8Ryw4lrZm8msj+vffEZUYyrW
...
ibpz9InSJoW3F7UcZzQvODt6EIuNlHDtzpljqLcr7iNIhpyR1J6nFo55WHQVDxLV
B39McBmmEdI==hI/9
-----END PGP MESSAGE-----
```

Por su parte, Esteban va a descifrar y comprobar el mensaje de la manera siguiente:

```
esteban@slyserver:~> gpg -d -v test.msg.asc
Version: GnuPG v2.0.9 (GNU/Linux)
gpg: encabezamiento de armadura:
gpg: la clave pública es ABC8FBE2
gpg: utilización de la subclave ABC8FBE2 en lugar de la clave
principal 149076EC

Necesita una frase de contraseña para desbloquear la
```

```
clave secreta para el usuario: «Esteban (Clave de prueba) <esteban@tele2.es>»
gpg: utilización de la subclave ABC8FBE2 en lugar de la clave
principal 149076EC
clave de 2048 bits ELG, ID ABC8FBE2, creada el 2009-05-06 (ID clave
principal 149076EC)

gpg: no running gpg-agent - starting one
gpg: cifrado con una clave de 2048 bits ELG, ID ABC8FBE2, creada el 2009-05-06
      «Esteban (Clave de prueba) <esteban@tele2.es>»
gpg: datos cifrados con AES256
gpg: nombre de archivo original: 'test.msg'
Hola, esto es un mensaje cifrado y encriptado
gpg: Firma
hecha el miérc. 06 de mayo de 2009 22:33:47 CEST con la clave DSA ID 13E021A8
gpg: utilización del modelo de confianza PGP
gpg: Firma correcta de «Sébastien ROHAUT (Slyce)
<sebastien.rohaut@dominio.es>»
gpg: firma binaria, algoritmo SHA1
```

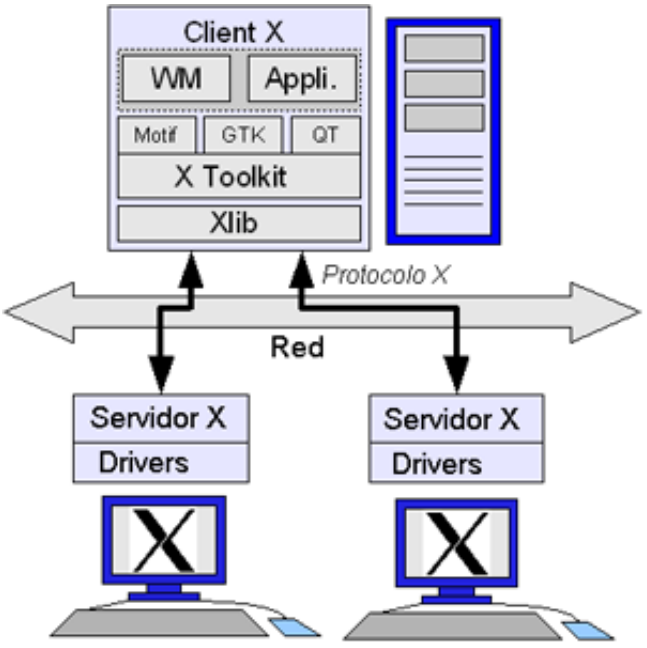
¿Cómo funciona un entorno gráfico?

1. X Window System

a. Un modelo cliente/servidor

Mientras en otros sistemas operativos se integra la interfaz gráfica en lo más profundo del sistema, Unix y Linux disponen de una arquitectura gráfica totalmente diferente. El sistema gráfico básico se llama **X Window System** o, de manera más corriente, **X Window**, **X11** o sencillamente **X**.

X no es sólo un simple programa. Es un sistema gráfico completo encargado de dibujar y gestionar los eventos de los componentes habituales de un entorno gráfico de usuario **GUI** (*Graphical User Interface*): ventanas, botones, menús, listas, menús desplegables, casillas de selección, cursores de ratón, etc. Tome nota: X puede gestionar y mostrar estos componentes gráficos, pero no se encarga de instalarlos. X sólo gestiona las interacciones entre el hombre y la máquina.



Arquitectura X Window

X tiene otra particularidad: es cliente/servidor. A menudo, el servidor X es un componente de software en un ordenador que dispone de un teclado, un ratón y una pantalla. Recibe y responde a órdenes de visualización, o procedentes del teclado o ratón. El cliente X se conecta al servidor y le manda órdenes de visualización, peticiones de entrada a través del teclado o el estado del ratón. Dicho de otro modo, un cliente X es un programa capaz de dialogar con el servidor X. En realidad, un cliente X es un programa gráfico. Para poder comunicarse con el servidor, utiliza un componente llamado Xlib. El cliente y el servidor no siempre están en la misma máquina. El servidor que gestiona la visualización puede encontrarse en un primer ordenador y el programa gráfico en otro. Las órdenes, llamadas peticiones, entre el cliente y el servidor pasan por la red.

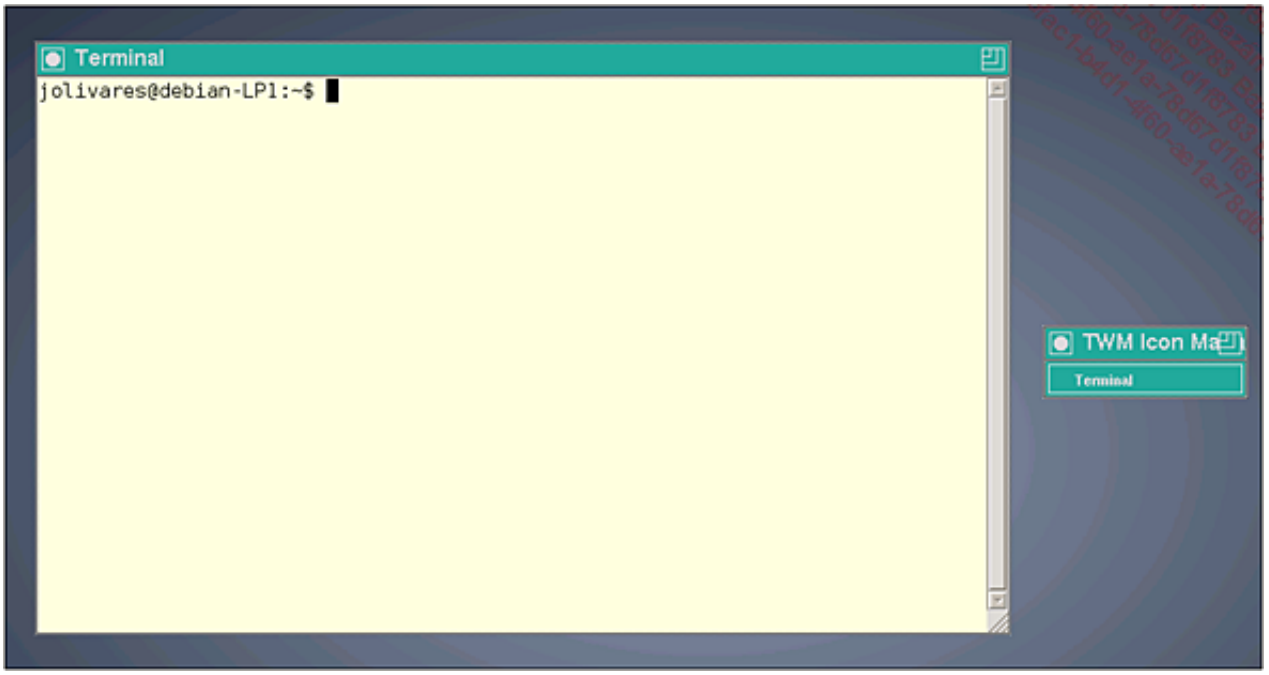
No confunda X Window con Windows. El primero es un sistema de visualización cliente/servidor; el segundo, un sistema operativo. El "Window" de X Window no tiene "s". Además, Windows y X Window no son compatibles entre sí, aunque se pueda instalar un servidor X en Windows.

Un usuario puede iniciar un programa (cliente) en su máquina local conectada por la red a un servidor remoto donde se está ejecutando realmente el servidor de ventanas.

Si ejecuta solo X Window, ejecuta únicamente el servidor. El resultado puede sorprenderle: una pantalla gris con una cruz como cursor de ratón. Por mucho que lo intente, no habrá ventana que venga a alegrarle la vista y los clics del ratón no tienen efecto alguno. Aunque iniciara un cliente, notaría que no hay ventanas.

b. El gestor de ventanas

Entre las muchas peticiones gráficas, la más importante es la de crear una ventana y decorarla dibujando los diversos elementos de ésta: la barra de título, el marco, los diversos botones. Como X sólo facilita lo imprescindible, se limita a dibujar la ventana, pero no le corresponde determinar cómo se debe renderizar el resto de los elementos. Otro programa cliente X debe decir al servidor cómo dibujar la ventana: es el **gestor de ventanas** o *Window Manager*. El servidor X visualiza el resultado dibujado por este gestor: ventanas, selecciones, movimientos y decoraciones (estilos, colores, etc.).



Un gestor simple: twm

Esto quiere decir que no hay un único gestor de ventanas, sino varios. Algunos son muy sencillos y básicos, y se limitan a lo mínimo; por ejemplo TWM, que aquí mostramos con una ventana y un menú. Observe su austeridad espartana. Sin duda, no es la mejor manera de sorprender a sus amigos con la calidad gráfica de Linux.

Otros son muy completos y permiten trabajar muy cómodamente porque, además de las ventanas básicas, proponen temas visuales agradables y personalizables, menús contextuales y, a veces, paneles de configuración. Es el caso de WindowMaker, con el que incluso se pueden cambiar menús, temas visuales, añadir botones en la barra de la derecha, etc. Durante mucho tiempo, este gestor de ventanas fue el más utilizado: rápido, eficaz, frugal en el consumo de recursos.

WindowMaker un gestor de ventanas evolucionado

c. Los widgets y los toolkits

Los componentes y su librería

Incluso a los gestores de ventanas más desarrollados les falta algo: una integración más avanzada de los programas (y de sus estilos) en el entorno. La razón estriba en el hecho de que, salvo casos muy especiales, los **WM** (*Windows Manager*) usan librerías gráficas propias con funcionalidades y componentes específicos. A los elementos que forman parte de la interfaz gráfica (botón, menú, campo de entrada, etc.) se les conoce como **Widget** (*Window gadget*), o sea, "chisme" para ventanas. Las librerías gráficas contienen juegos completos.

Todas estas librerías gráficas deben ofrecer funciones para dibujar y gestionar widgets. Cuando la librería contiene un kit completo y extendido de widget, se llama **Widget Toolkit**, o de manera más sencilla **toolkit** o caja de herramientas para la interfaz gráfica.

Existen varios, que se diferencian según su complejidad, su uso, su estética (relativamente), el lenguaje de programación para utilizarlos, etc. X Window dispone de un toolkit por defecto llamado**Xt** (*X Toolkit*). El más conocido y utilizado ha sido durante mucho tiempo **MOTIF**, más aún porque es un estándar POSIX IEEE, hasta que, liberado en torno al año 2000, MOTIF fue considerado software propietario y, por esa razón, los programadores gráficos crearon otros conjuntos de componentes. La consecuencia es de lo más interesante: una pléyade de toolkits diferentes, incompatibles entre sí, que dibujan de manera totalmente diferente los widgets. Todavía puede encontrar en Linux muchos ejemplos en los que un componente aparece visualmente diferente en varios programas, dando la impresión de caos de estilos heterogéneos.

GTK, Qt

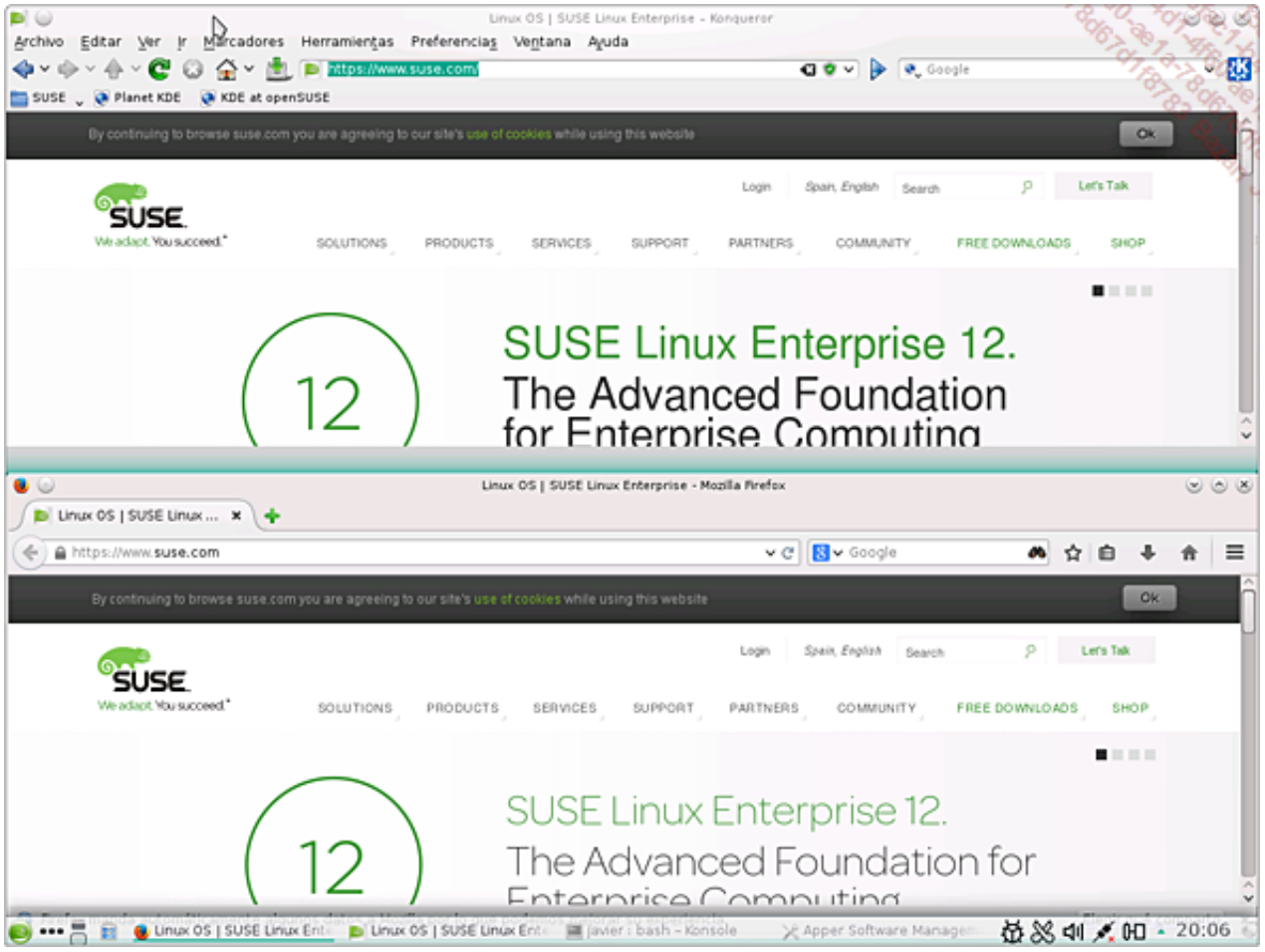
Al no poder utilizar Motif (y su equivalente libre Lesstif, que no estaba todavía muy preparado), los

desarrolladores de interfaces gráficas en Linux concentraron sus esfuerzos en dos toolkits concretos, excelentemente programados y que permiten hacer abstracción totalmente de X Window. Existen varias versiones que no necesitan siquiera X Window (para Windows, Macintosh, los asistentes personales, los teléfonos, etc.).

- El primero se llama **GTK** (*GIMP toolkit*). Fue creado para dibujar la interfaz gráfica del programa de retoque de imágenes GIMP. Gracias a su eficacia gráfica, muchos programadores lo han incorporado a sus propios programas y GTK con el tiempo se ha vuelto independiente. Hoy es la librería por defecto del entorno de escritorio GNOME. GTK está escrito en lenguaje C.
- El segundo se llama **QT** (*cute*, *mono*). La Q no significa nada: los desarrolladores de este toolkit pensaban que esta consonante, Q, era muy bonita en el editor de textos que utilizaban... Está desarrollado en C++, pero es compatible con una decena de lenguajes. Además de las funciones de dibujo y gestión de los widgets, QT propone un entorno completo de desarrollo de aplicaciones gráficas y no gráficas: bases de datos SQL, XML, multithreading, gestión de archivos, internacionalización, etc. La librería QT es la utilizada por el entorno ofimático KDE.
- Hay otros muchos toolkits, como **MOTIF**, **AWT**, **ATHENA**. Aunque son menos conocidos, todavía hay aplicaciones que los utilizan.

GTK y QT no son compatibles, al menos de manera nativa. Si ejecuta un programa desarrollado con GTK en KDE, el contorno de la ventana tendrá el estilo de KDE (QT), pero GTK dibujará su contenido. Mire el resultado en la captura de pantalla siguiente. La ventana de arriba representa el navegador de Internet Firefox, que utiliza la librería GTK, mientras que la ventana de abajo representa el navegador Konqueror, que utiliza la librería QT. Los menús, los botones y el contenido no son idénticos. Afortunadamente, SUSE Linux hace bien las cosas y propone una homogeneización de los temas gráficos. En su parte final, este capítulo le explica cómo optimizar la visualización de sus temas gráficos.

Algunos programas utilizan sus propios toolkits. Es el caso de OpenOffice.org y es la principal razón de su lentitud durante su primera descarga. Cuando ejecuta este programa, éste debe cargar también su toolkit, lo cual resulta redundante con el de su gestor de ventanas o de su entorno ofimático. Afortunadamente, las últimas versiones se adaptan a su entorno, pero algunos programas (o más bien sus programadores) aún se resisten.



Dos juegos de herramientas, dos estilos

d. Los escritorios virtuales

Si hay una característica de Linux que se echa de menos en sus competidores, es la del escritorio virtual. ¿Qué hacer cuando hay tantas ventanas abiertas que estamos obligados a minimizarlas para recuperar las escondidas, o volver a dimensionarlas para verlas? Pues utilizar los escritorios virtuales.

X Window permite la utilización de varios espacios de trabajo en una misma sesión. En vez de tener un escritorio, puede tener dos, tres, cuatro... En efecto, en teoría puede obtener tantos escritorios como desee si dispone de bastante memoria. Es el gestor de ventanas, o el entorno de escritorio, el que elige (y por lo tanto, usted). KDE permite, por ejemplo, disponer de 20 escritorios.

El interés es obvio. En vez de concentrar todas sus ventanas en un solo escritorio, repártalas en los escritorios disponibles. Más aún: sólo precisa de un simple clic para ir de un escritorio a otro, o incluso para desplazar una ventana de un escritorio a otro. Además, es posible hacer que una ventana se siga visualizando en varios escritorios virtuales (es decir, "colgarla" en la pantalla).

2. Los entornos de escritorio

Si Unix, y Linux en particular, han conservado durante mucho tiempo la fama de sistema operativo para profesionales, universitarios, informáticos o excéntricos gurús, es porque faltaba un entorno gráfico orientado a la ofimática y de fácil manejo. Ya en los primeros capítulos se comentaba que los hábitos que los usuarios han ido adquiriendo con las interfaces de Windows o MacOS no pueden desaparecer de un día a para otro. Durante años, los gestores de ventanas fueron difíciles de configurar. ¿Se le puede pedir a alguien acostumbrado al ratón que modifique un archivo de configuración en modo texto, cuya sintaxis no es precisamente sencilla?

Hacia 1996, Linux alcanzó la madurez entre las empresas, pero sigue habiendo un mercado por

conquistar. En el otoño de 1995, Windows puso un contundente final a la carrera de la mayoría de los sistemas operativos para PC de oficina. Los DR-DOS, PC-DOS y sobre todo OS/2, muy superiores, han sobrevivido primero y, finalmente, desaparecido. Los entornos gráficos de Unix y Linux no pueden competir en términos de facilidad de manejo y oferta de programas para el gran público con los sistemas de Microsoft y de Apple. Todo queda por hacer: hay que alcanzar a los competidores, mejorar e innovar.

Este titánico trabajo no desanima a los programadores. Ya que Linux, para el gran público, no es de fácil manejo, hay que crear un entorno ofimático gráfico adaptado a sus necesidades. Con pocos meses de diferencia dos equipos distintos asumen el proyecto.

KDE

Logo de KDE

En octubre de 1996 un primer equipo liderado por Matthias Ettrich anuncia el lanzamiento del proyecto **KDE** (*K Desktop Environment*). La K ya no tiene significado particular (salvo al principio: kool) y sólo es la letra más cercana a la L de Linux en el alfabeto. KDE ofrece una interfaz gráfica y aplicaciones unificadas alrededor de un toolkit llamado QT y desarrolladas en C++. KDE 1 ya era impresionante en 1997 y permitía utilizar Linux como si fuera una mezcla de Windows 3.1 y 95. Pero fue a partir de KDE 2, ya en el año 2000, y gracias a una nueva arquitectura, cuando KDE reveló su potencia e igualó a los entornos gráficos de sus competidores. La salida de la versión 3 (la versión 3.5.9 data de febrero de 2008) va aún más lejos y, superando a todos sus competidores, integra los diversos componentes de forma avanzada. KDE es el entorno gráfico favorito de la mayoría de los usuarios de Linux; pretende ser el más completo en términos de integración, de configuración y de oferta de software (a riesgo de tener numerosos duplicados y dejar la opción al usuario). KDE también es el entorno de escritorio utilizado por Linus Torvalds, el creador de Linux.

La versión 4.0, publicada a principios de 2008, supone una remodelación total del entorno ofimático KDE. A diferencia de la evolución constante y continua entre las versiones 1.0 y 3.x, KDE 4 representa una ruptura total en relación con sus antecesores. La principal novedad son componentes llamados ”plasmoides”. De hecho, la ruptura es tan total que la versión 4.0 considerada como estable no lo es tanto en el momento de escribir este capítulo. La versión 4.1, anunciada en julio de 2008, es la primera que se puede utilizar en un entorno de producción. La versión 4.14 salió en julio de 2014. Se han llevado a cabo grandes esfuerzos de integración y KDE no tiene nada que ver con las interfaces de Windows y de Mac OS X.

KDE se sigue conociendo bajo ese nombre, pero hoy en día se presenta como una plataforma llamada a su vez Plataforma KDE.

GNOME

Logo de Gnome

En agosto de 1997, un equipo liderado por Miguel de Icaza y Federico Mena decide crear un entorno

de escritorio totalmente libre para competir con KDE (cuyo QT no era libre en aquel momento).**GNOME** (*Gnu Network Object Model Environment*) es un entorno de escritorio basado en el toolkit GTK y programado en C. Es el entorno gráfico de escritorio oficial del proyecto GNU. La filosofía de GNOME difiere de manera radical de la de KDE. GNOME tiene un estilo voluntariamente depurado, que se centra en las funcionalidades esenciales de un entorno y que privilegia algunas aplicaciones en detrimento de otras. La integración no es tan exhaustiva, ya que las aplicaciones desarrolladas en GTK no son todas aplicaciones GNOME. GNOME 3, un remake completo, salió en abril de 2011. La última versión de julio de 2014 es la 3.12.

El proyecto Freedesktop

Logo de Freedesktop

Los desarrolladores de KDE y GNOME decidieron trabajar en una base común de configuración para una mejor integración de las aplicaciones GNOME en KDE, y viceversa, con el fin de armonizar la infraestructura del conjunto. El objetivo no es diseñar un escritorio único, ya que cada uno tiene sus ventajas. El resultado es la creación de **Freedesktop**, una zona de comunicación y colaboración informales destinada a trabajar en favor de la interoperabilidad de los diversos entornos gráficos para Linux y Unix. Freedesktop es neutral a nivel tecnológico, no se privilegia ningún entorno. Pero, a menudo, lo que sale de Freedesktop se integra en GNOME y KDE: del menú principal común al programa de detección de nuevo hardware, pasando por los copiar y pegar, los atajos de teclado y una multitud de mejoras que Freedesktop ha aportado.

Xorg

1. Presentación

Desde su aparición hasta 2004, Linux (así como las distribuciones BSD) iba acompañado del entorno X Window libre XFree86. Un cambio de licencia que hizo que XFree86 fuera un poco menos libre y en particular que dejara de ser compatible con la GPL de la Free Software Foundation, tuvo como consecuencia un ”fork” (creación de una nueva rama de desarrollo) a partir de la última versión con licencia MIT.

X.org termina por agrupar la mayoría de los antiguos desarrolladores de XFree86. X.org era un *fork*de XFree86 con fecha anterior al cambio de licencia, pero con ideas innovadoras bastante rechazadas por los partidarios de XFree86. X.org, llamado de manera común Xorg, se convierte en el entorno X Window principal de Linux y aún hoy lo sigue siendo.

X.org es rápido, modulable, está disponible con numerosos drivers, compatible. Las últimas versiones son capaces de la autodetección del hardware sin necesidad de complejos archivos de configuración.

Logo Xorg

La última versión de Xorg, de julio de 2014 es la 1.16.0 y se basa en la versión 7.7 de X11.

2. Instalación

Se entrega Xorg con todas las distribuciones de Linux. Las distribuciones del primer semestre de 2014 se suelen suministrar con Xorg 1.14. Resulta que el desarrollo y las innovaciones son tan rápidos que a veces les cuesta seguir el ritmo a las herramientas asociadas (en particular para la configuración).

Puede recuperar el código fuente de X.org, pero debe saber que le esperan horas y horas de compilación. Es mejor instalar Xorg desde los CD, DVD o repositorios de su distribución.

Por defecto, se instala Xorg en /usr/X11R6. Sin embargo, algunas distribuciones colocan ahora los binarios, las librerías, los archivos compartidos, etc., en el árbol clásico /usr y en particular los módulos y drivers en /usr/lib/xorg (o /usr/lib64/xorg).

En cambio, la configuración se encuentra siempre en el mismo sitio: /etc/X11.

A continuación, le presentamos un ejemplo de paquetes de Xorg instalados en una distribución Fedora 20 de 64 bits:

```
# rpm -qa|grep xorg
xorg-x11-glamor-0.5.1-3.20140115gitfb4d046c.fc20.x86_64
xorg-x11-drv-evdev-2.8.4-1.fc20.x86_64
xorg-x11-drv-mga-1.6.2-8.fc20.x86_64
xorg-x11-drv-synaptics-1.7.6-2.fc20.x86_64
xorg-x11-xkb-utils-7.7-8.fc20.x86_64
xorg-x11-font-utils-7.5-18.fc20.x86_64
xorg-x11-drv-intel-2.21.15-7.fc20.x86_64
xorg-x11-drv-1.0.9-2.fc20.x86_64
abrt-addon-xorg-2.2.1-2.fc20.x86_64
xorg-x11-utils-7.5-12.fc20.x86_64
```

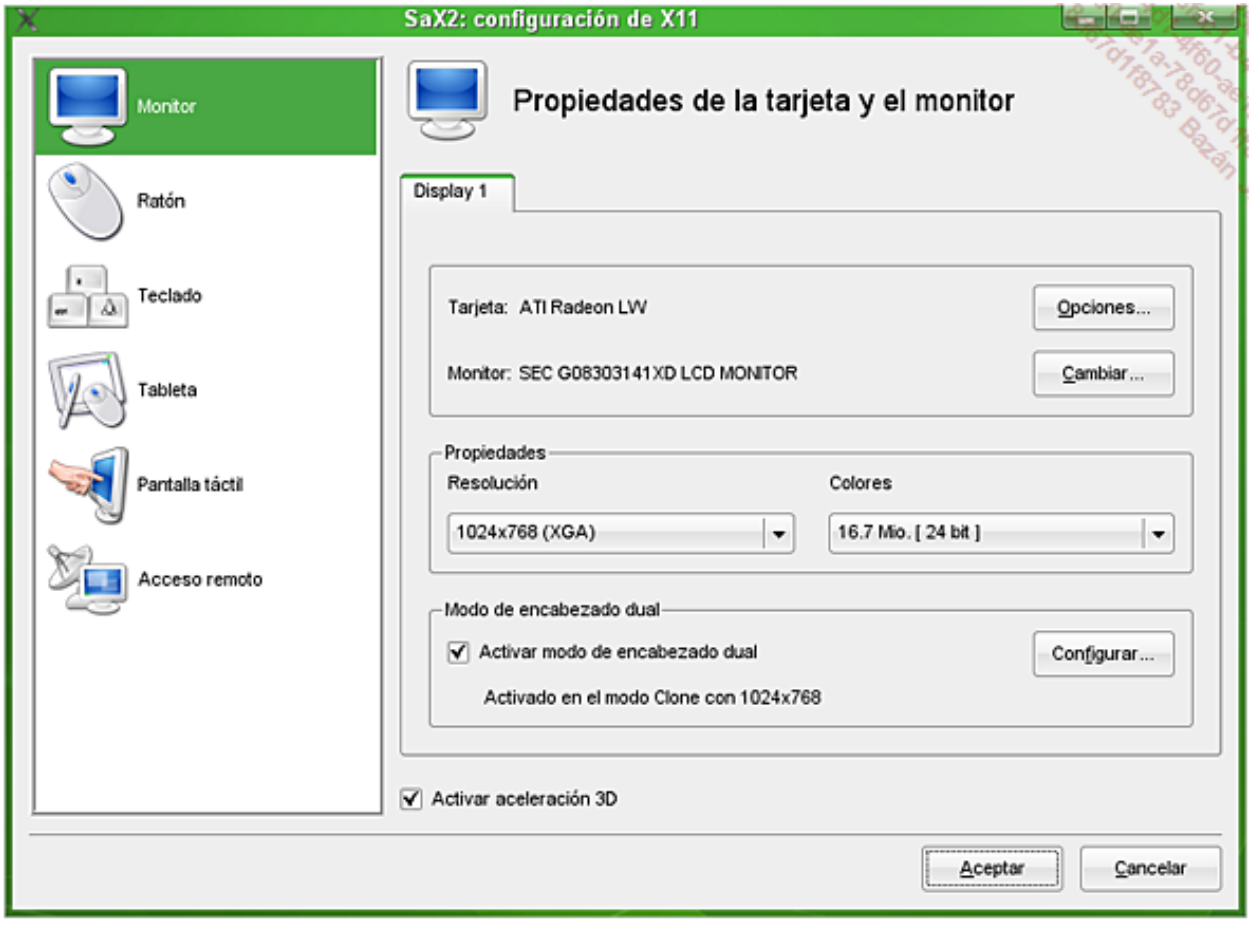
```
xorg-x11-server-common-1.14.4-11.fc20.x86_64
xorg-x11-xauth-1.0.9-1.fc20.x86_64
xorg-x11-drv-modesetting-0.8.0-2.fc20.x86_64
xorg-x11-drv-wacom-0.23.0-5.fc20.x86_64
xorg-x11-drv-vmmouse-13.0.0-6.fc20.x86_64
xorg-x11-server-Xorg-1.14.4-11.fc20.x86_64
xorg-x11-server-utils-7.7-6.fc20.x86_64
xorg-x11-drv-vmware-13.0.2-4.20140613git82c9b0c.fc20.x86_64
xorg-x11-drv-vesa-2.3.2-10.fc20.x86_64
xorg-x11-drv-fbdev-0.4.3-10.fc20.x86_64
xorg-x11-xinit-1.3.2-9.fc20.x86_64
xorg-x11-drv-ati-7.2.0-3.20131101git3b38701.fc20.x86_64
xorg-x11-drv-qxl-0.1.1-3.fc20.x86_64
xorg-x11-drv-openchrome-0.3.3-2.fc20.x86_64
```

3. Configuración

a. Por distribución

El archivo de configuración de Xorg está en /etc/X11 y se llama xorg.conf: /etc/X11/xorg.conf. No es habitual, pero sí posible, crear un archivo xorg.conf completamente a mano. En la práctica, la fuerza de las distribuciones Linux reside en parte en su capacidad en detectar su hardware y configurar el entorno gráfico en consecuencia. De hecho, son estas herramientas las que definen, según sus indicaciones y el hardware detectado, el archivo xorg.conf.

En openSUSE, la herramienta SaX permite configurar el entorno gráfico.



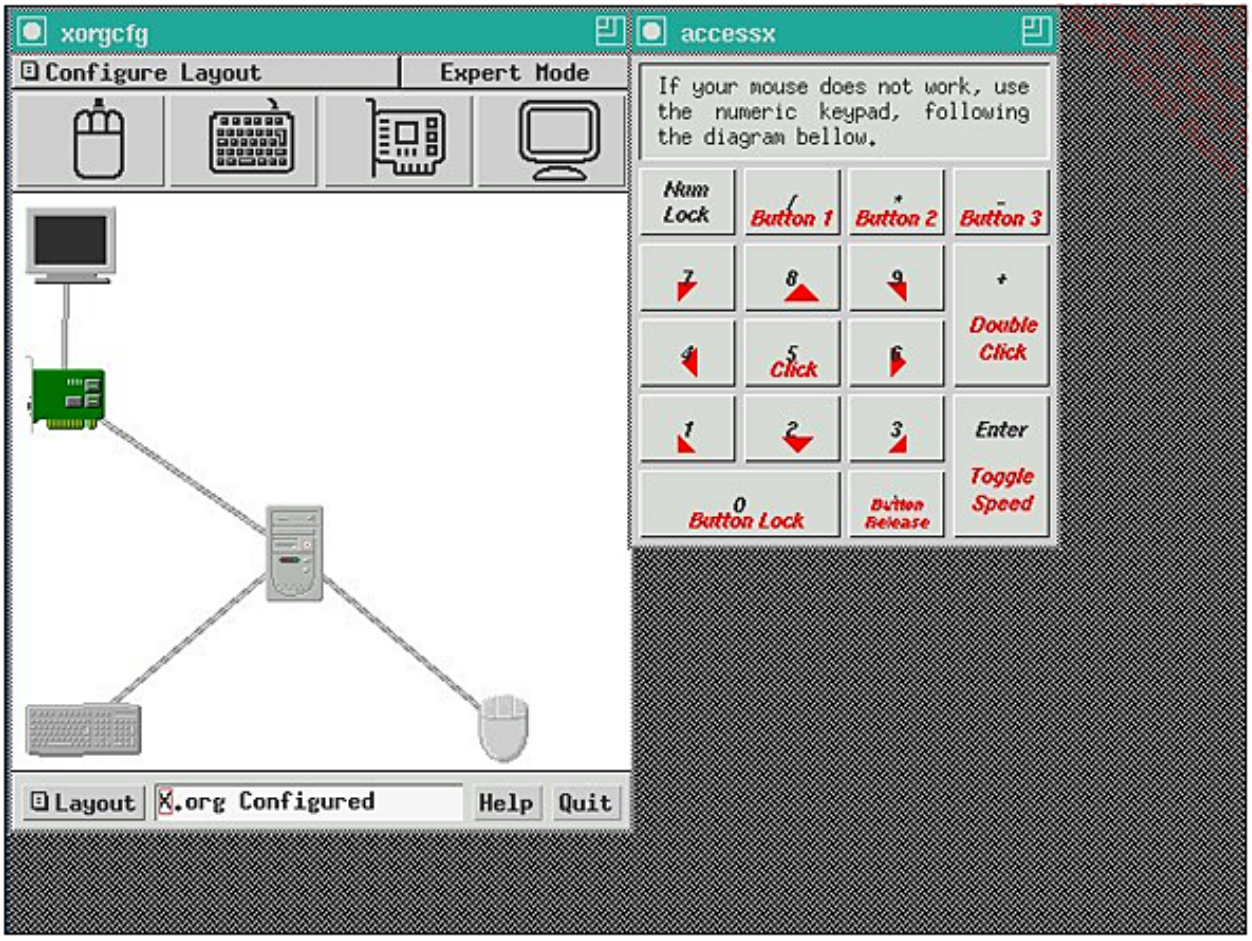
La herramienta de configuración X Sax 2 de openSUSE

Las últimas versiones de Xorg son autoconfigurables. El servidor X intenta detectar automáticamente su hardware incluso si no hay archivo de configuración. De este modo funciona generalmente bien. Sin embargo, siempre puede modificar usted mismo el archivo de configuración.

b. Xorgcfg

El comando **xorgcfg** intenta detectar de forma automática todos los parámetros de configuración del servidor X y crear un archivo xorg.conf. En la práctica, una vez que el servidor X dispone de una configuración básica, se inicia X con un gestor de ventanas básico (twm) y una herramienta gráfica que permite modificar, o más bien ajustar, la configuración.

Recuerde que tratamos con una herramienta básica y que luego será preciso modificar la configuración a mano en el archivo resultante. Además, las pruebas realizadas durante la redacción de esta parte del libro han mostrado que xorgcfg es inestable: si "mata" Xorg en este momento (con [Alt][Ctrl][Supr]), el ordenador parece reaccionar mal, y quizá haya que reiniciarlo con el botón reset.



Xorgcfg, herramienta de configuración básica

c. Xorgconfig

La herramienta **xorgconfig** es similar a xorgcfg, pero está basada en un asistente en modo cuestionario. Equivale y sustituye a la herramienta **xf86config** del antiguo servidor X de XFree86. Tiene que contestar, en orden, a las preguntas siguientes:

- ¿Cuál es el protocolo de comunicación con el ratón?
- ¿Desea una emulación de los tres botones con los otros dos?
- ¿Cuál es el periférico del ratón?
- ¿Cuál es el tipo de teclado?
- ¿En qué idioma?
- ¿Tiene opciones particulares?
- ¿Cuál es su modelo de pantalla (tipo genérico)?

- ¿Cuáles son las frecuencias de barrido de su monitor?
- ¿De cuánta memoria dispone su tarjeta gráfica?
- ¿Cuál es la profundidad de los colores (por ejemplo, 24 bits)?
- ¿Dónde guardar el archivo?

Durante mucho tiempo, este tipo de herramienta era la única disponible para configurar el servidor X. Ahora, las herramientas de las distribuciones lo hacen muy bien, hasta el punto de que el archivo de configuración no se encuentra frecuentemente, ya que es poco útil. Sin embargo, a pesar de todo debe conocer las bases de la escritura del archivo de configuración xorg.conf: las modificaciones manuales pueden ser habituales en este archivo.

d. X

El propio servidor X puede implementar una configuración básica escribiendo el comando siguiente desde una consola de texto (no debe haber ningún servidor X ejecutándose previamente).

```
# X -configure :0
```

4. Estructura de xorg.conf

a. División

El archivo **`/etc/X11/xorg.conf`** se ordena en forma de secciones y de subsecciones. Cada una corresponde o a una funcionalidad del servidor X o a un periférico de entrada o salida.

Para hacer funcionar X, hace falta:

- una salida, una pantalla y la tarjeta gráfica asociada;
- una entrada, teclado, ratón, etc.

En este último caso, la carencia de ratón podría impedir el arranque de todo el sistema. A continuación se detalla el contenido de las diferentes secciones.

b. Valores booleanos

Determinadas opciones de xorg.conf aceptan valores de texto o numéricos, pero muchas sólo toman dos valores. En este caso, son posibles varias opciones:

- Se consideran como VERDADERO los valores 1, on, true o yes.
- Los valores considerados como FALSO serían 0, off, false y no.

c. Sección InputDevice

Una sección **InputDevice** describe un periférico de entrada:

- teclados,
- ratones,
- touchpads,
- pantallas táctiles,
- etc.

Cada periférico de entrada cuenta con su propia sección.

Una sección InputDevice se compone de:

- Un identificador único, **Identifier**.
- Un controlador, **Driver** (p. ej.: kbd, mouse, evdev, etc.).
- Las diversas opciones, **Option**, relacionadas con el driver.

Las rutas de los archivos de periféricos se suelen colocar en la sección **Files**, bajo las entradas**InputDevice**.

A continuación le presentamos el ejemplo de la sección que describe un teclado. El driver se llama**kbd**. La sección cuenta con varias opciones que sirven para describir el modelo, el tipo (azerty, qwerty, etc.), el número de teclas, opciones adicionales, etc. Aquí se describe un teclado español de 105 teclas.

```
Section "InputDevice"
  Identifier   "Keyboard[0]"
  Driver       "kbd"
  Option       "Protocol" "Standard"
  Option       "XkbLayout" "es"
  Option       "XkbModel" "pc105"
  Option       "XkbOptions" "caps:shiftlock"
  Option       "XkbRules" "xfree86"
EndSection
```

Encontrará ayuda en las manpages, bajo el nombre kbd:

- En Linux el protocolo es siempre **Standard**.
- **XkbModel** define el modelo de teclado. Los teclados recientes toman como valor **pc105**, incluso los teclados con teclas multimedia.
- **XkbLayout** define el idioma del teclado, **es** para un teclado español, **us** para uno estadounidense, **be** para un belga, etc.
- **XkbOptions** facilita opciones específicas. Por ejemplo, **caps:shiftlock** permite obtener el mismo funcionamiento que un teclado en Windows: la tecla CapsLock en Linux permite acceder en principio a las mayúsculas acentuadas, con esta opción accede a las cifras.

Veamos ahora una sección que corresponde al ratón (un modelo de Logitech):

```
Section "InputDevice"
  Identifier   "Mouse[1]"
  Driver       "evdev"
  Option       "Buttons" "10"
  Option       "InputFashion" "Mouse"
  Option       "Name" "Logitech USB R*"
  Option       "Protocol" "event"
  Option       "SendCoreEvents" "on"
  Option       "Vendor" "Sysp"
  Option       "ZAxisMapping" "4 5"
EndSection
```

Podrá observar que la estructura sintáctica es idéntica; sólo cambian el driver y las opciones. Si bien X reconoce la mayoría de los ratones, a menudo hace falta modificar las opciones **Buttons** y**ZaxisMapping**. Además, la interpretación de los eventos de los botones depende de las aplicaciones, entornos de escritorios, etc., y no de X Window.

El driver se llama aquí **evdev**. Un driver genérico es el encargado de gestionar todo tipo de eventos en entrada, tanto para los teclados como los ratones, y sobre todo para estos últimos, ya que es más complicado instalarlo para los teclados. Pero existen drivers específicos: **kbd** es uno de ellos; **mouse**, otro. En cuanto a los dispositivos portátiles, se utiliza a menudo el driver **synaptics**para los touchpads, e incluso el driver **wacom** para los tabletPC y las tabletas gráficas del mismo nombre. A continuación, se muestra una entrada encontrada en un eeePC de

Asus:

```
Section "InputDevice"
    Identifier "Touchpad"
    Driver     "synaptics"
    Option     "SHMConfig" "on"
EndSection
```

d. Sección Monitor

La sección **Monitor** describe la pantalla de su ordenador. Hoy en día todos los monitores son de tipo DDC, es decir, son capaces de informar mediante el driver gráfico, y por lo tanto al servidor X, de las frecuencias y las resoluciones que soportan. Por ello la sección **Monitor** se ve reducida a la mínima expresión.

Sin embargo, si el modelo no es compatible (lo que puede ocurrir teniendo en cuenta el driver gráfico) o si el monitor devuelve información poco fiable, puede elegir un modelo genérico, o definir por sí mismo las opciones de su monitor. Para ello, utilice la documentación de éste y busque en particular:

- El tamaño real de la visualización, anchura y altura en milímetros.
- El rango de frecuencias horizontales.
- El rango de frecuencias verticales.
- Si es preciso la lista de las resoluciones soportadas.

La sección del siguiente ejemplo describe un monitor genérico de tipo Vesa que soporta una resolución de 1280x1024 a 60 Hz. Funciona perfectamente con la pantalla LCD de 19 pulgadas 4/3 del autor.

```
Section "Monitor"
    Identifier    "Monitor[0]"
    VendorName    "--> VESA"
    ModelName     "1280X1024@60HZ"
    UseModes      "Modes[0]"
    DisplaySize   340 270
    HorizSync     31.0 - 64.0
    VertRefresh   50.0 - 60.0
    Option        "DPMS"
EndSection
```

Si su monitor informara de sus frecuencias y de las resoluciones soportadas, y el driver asociado supiera interpretarlas, no necesitaría toda esta descripción. La siguiente sección funciona también muy bien, ya que el driver y el monitor están en la misma onda.

```
Section "Monitor"
    Identifier   "L1915S"
    ModelName    "L1915S"
    VendorName   "LG"
EndSection
```

La mayoría de las opciones hablan por sí mismas. No obstante, le recomendamos que tome buena nota del siguiente punto: UseModes.

e. Sección Modes

La sección **Modes** o **UseModes** está asociada comúnmente a la sección Monitor. Algunos monitores, y no siempre antiguos o sin marca, necesitan ajustes específicos de frecuencias y de timings para acceder a determinadas resoluciones de visualización. En Windows, estos ajustes son totalmente transparentes para el usuario: o bien el driver es genérico (como los ajustes del primer monitor del

punto anterior), o bien la configuración se encuentra en el archivo de descripción ".inf" facilitado con el monitor.

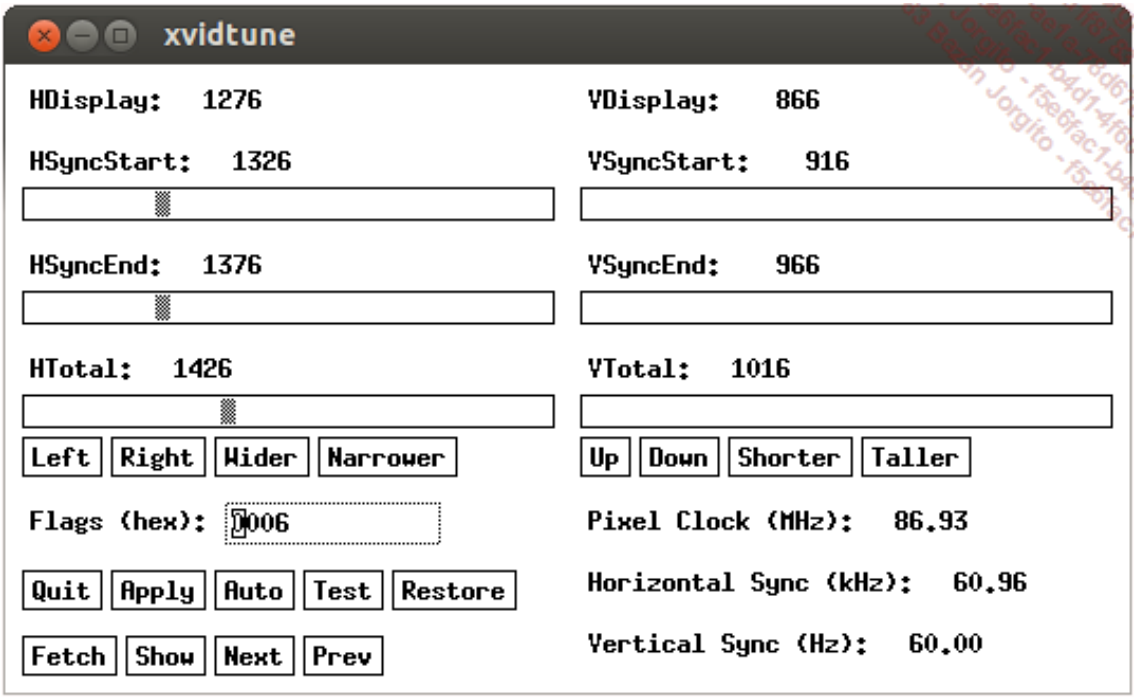
Los monitores recientes (desde hace unos años) no suelen necesitar este tipo de configuración.

Xorg y la mayoría de las herramientas de configuración disponen de una base de datos de monitores bastante completa, pero es posible que el suyo no esté en ella.

```
Section "Modes"
  Identifier    "Modes[0]"
  ModeLine      "1280x1024" 102.6 1280 1312 1472 1632 1024 1028
1032 1048 -hsync -vsync
EndSection
```

No se le pedirá crear *modelines*. Puede encontrar ayuda en las direcciones siguientes:

- http://www.x.org/wiki/FAQVideoModes le explica particularmente cómo crear, mediante el registro de eventos de xorg, sus propios *modelines* según la resolución deseada. Aún hace falta que el driver facilite esta información.
- http://xtiming.sourceforge.net/cgi-bin/xtiming.pl le permite calcular *modelines* según las especificaciones de su monitor. Puede hacer milagros, pero es arriesgado utilizarlo.
- http://john.fremlin.de/programs/linux/read-edid/ le proporciona programas que permiten descodificar la información EDID de su monitor.
- http://gentoo-wiki.info/TIP_Getting_modelines le explica cómo interpretar y utilizar los resultados del comando anterior.



xvidtune permite ajustar de manera muy fina las resoluciones

Si su monitor parece que no funciona con una resolución que se supone que debería aceptar (pantalla negra, de espera, X que sale con el mensaje "no screen found", ninguna resolución aplicable, etc.), piense en comprobar dentro del archivo xorg.conf si la sección Monitor hace llamada a modelines; aísle la línea correspondiente y coméntela. Vuelva a iniciar X ([Alt][Ctrl][Supr]): puede que sea el origen del problema.

Apoyándose en la configuración de X, podría intentar optimizar la visualización de su pantalla. Los casos clásicos (para pantallas CRT, con tubo) son una visualización excéntrica, ni demasiado alta o ancha, o lo contrario. Si los ajustes de su monitor no permiten obtener una buena imagen, puede ayudarse con la herramienta **xvidtune**, que permite influir sobre los ajustes de los modos de vídeo.

¡Cuidado! **xvidtune** puede hacer milagros, pero también puede deteriorar de manera irrevocable su pantalla si indica valores de timings y frecuencias estrafalarios. Use esta herramienta con cuidado.

f. Sección Device

La sección **Device** describe el driver y las opciones de la tarjeta gráfica. Como puede haber varias tarjetas gráficas, puede haber varias secciones de este tipo, con identificadores diferentes. Tenga en cuenta que, con determinados drivers, algunas opciones se colocan en otra sección, **Screen**. El ejemplo siguiente muestra la configuración de una tarjeta gráfica NVidia con el driver propietario.

```
Section "Device"
  Identifier    "Device[0]"
  Driver        "nvidia"
  VendorName    "NVidia"
  BoardName     "GeForce 8600 GT"
  Option        "NoLogo" "0"
  Option        "DPI" "86 x 86"
  Option        "RenderAccel" "True"
  Option        "AddARGBGLXVisuals" "True"
EndSection
```

Al igual que los demás drivers, las opciones dependen del driver y no son siempre las mismas entre uno y otro. Las opciones mostradas antes son propias del driver Nvidia. La primera permite evitar o no la visualización del logo del fabricante al inicio y la segunda fuerza el número de DPI de la visualización. Las dos últimas activan la aceleración del display (en el ejemplo, aceleración mixta) y algunas optimizaciones para las extensiones OpenGL.

Aquí tiene un segundo ejemplo con una tarjeta ATI y el driver fglrx propietario:

```
Section "Device"
  Identifier    "aticonfig-Device[0]"
  Driver        "fglrx"
  Option        "XAANoOffscreenPixmaps" "true"
  Option        "TexturedVideo" "On"
  Option        "UseFastTLS" "1"
  Option        "Textured2D" "on"
  Option        "TexturedXRender" "on"
  Option        "BackingStore" "on"
  Option        "VideoOverlay" "Off"
  Option        "OpenGLOverlay" "Off"
  BusID         "PCI:1:0:0"
EndSection
```

Para saber más cada una de las opciones, puede remitirse al manual de uso del driver.

g. Sección Screen

La sección **Screen** es una especie de metasección: llama a las secciones Monitor y Device para agrupar todos los parámetros de visualización. Conociendo las capacidades tanto de la tarjeta gráfica como del monitor, es posible elegir qué resoluciones gráficas deben estar accesibles, para qué número de colores y cuáles son los modos por defecto.

Una sección **Screen** contiene una o varias subsecciones llamadas **Display** que determinan, para un

tipo de visualización en n bits (8 bits: 256 colores, 16 bits: 65.536 colores, 24 bits: 16 millones de colores), cuáles deberían ser las resoluciones adecuadas.

- **Depth** define en número de bits la profundidad de los colores.
- **Modes**, las resoluciones soportadas.
- La primera resolución de la lista es la resolución por defecto.
- Puede pasar de una resolución a otra con [Ctrl][Alt] + o - o mediante las posibilidades propuestas por su entorno de escritorio.
- Si no se soporta una resolución, se desactiva automáticamente.

En el ejemplo siguiente, estándar, se han agrupado las secciones anteriores Device y Monitor. La sección Screen se llama Screen[0]. Cuatro subsecciones **Display** configuran la visualización en 8, 15, 16 y 24 bits. Cuatro resoluciones deberían ser accesibles para cada una, la resolución por defecto es de 1280x1024.

La entrada **DefaultDepth** precisa la profundidad de los colores por defecto. En este caso, la visualización por defecto será en 1.280x1.024 y 16 millones de colores.

```
Section "Screen"
  Identifier    "Screen[0]"
  Device        "Device[0]"
  Monitor       "Monitor[0]"
  DefaultDepth   24
  SubSection    "Display"
    Depth        15
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
    Depth        16
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
    Depth        24
    Modes       "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
  SubSection    "Display"
    Depth        8
    Modes        "1280x1024" "1024x768" "800x600" "640x480"
  EndSubSection
EndSection
```

h. Sección ServerLayout

Un archivo xorg.conf dispone de al menos una sección ServerLayout, y en general es la sección básica que agrupa las secciones de visualización y de entrada necesarias para el arranque de X Window y el funcionamiento de la sesión. Se compone de:

- un identificador único, Identifier;
- una o varias entradas **Screen** (en caso de multi-head, doble pantalla o más);
- una o varias entradas **InputDevice** (para el teclado, el ratón, las tabletas, etc.).

X gestiona muy bien las visualizaciones multipantalla y multitarjeta mediante una extensión llamada **Xinerama**. A continuación le presentamos un caso sencillo:

```
Section "ServerLayout"
  Identifier   "Layout[all]"
  Screen       "Screen[0]" 0 0
  InputDevice  "Keyboard[0]" "CoreKeyboard"
  InputDevice  "Mouse[1]" "CorePointer"
```

```
EndSection
```

Detengámonos en lo que sigue al nombre de las secciones **Screen** e **Inputdevice**:

- Los **0 0** después de **Screen** informan sobre posición de la pantalla en caso de modo dual o multi-head. Se trata de las posiciones X e Y partiendo de la esquina superior izquierda.
- El **CoreKeyboard** indica que este periférico de entrada es el teclado principal. Sólo puede haber uno. Si no hay, X busca todo tipo de periféricos de entrada que puedan actuar como tal.
- El **CorePointer** es lo mismo, pero para el puntero del ratón.

Si dispone de varios ratones (p. ej.: ratón USB y touchpad) o teclados (p. ej.: un teclado USB enchufado en la base de un portátil, y el teclado del portátil), se pueden declarar los otros periféricos de entradas en la sección ServerLayout, pero en este caso escriba "**SendCoreEvents**" y así sucesivamente: esto permite al periférico enviar eventos al servidor X como los otros. X podrá gestionarlos todos:

```
Section "ServerLayout"
  Identifier    "Layout[all]"
  Screen        "Screen[0]" 0 0
  InputDevice   "Keyboard[0]" "CoreKeyboard"
  InputDevice   "Mouse[1]" "CorePointer"
  InputDevice   "Touchpad" "SendCoreEvents"
EndSection
```

i. Sección Files

La sección **Files** indica al servidor X las rutas hacia algunos archivos o directorios que pueden serle necesarios:

- los periféricos de entrada;
- las fuentes de caracteres;
- los módulos complementarios;
- la base de los códigos colores RGB (un poco como los códigos HTML).

Algunas rutas no hay que establecerlas, ya que están definidas por defecto. De hecho, y en teoría desde la versión 7.3, no es necesaria ninguna ruta alguna, pero puede ser útil precisar otras desconocidas. Las entradas son las siguientes:

- **InputDevices**: la ruta de los periféricos de entrada;
- **FontPath**: la ruta de las fuentes de caracteres;
- **RGBPath**: la ruta del archivo de los códigos RGB;
- **ModulePath**: la ruta de los módulos complementarios de Xorg.

El ejemplo siguiente, precisa las rutas de los periféricos y de las fuentes.

```
Section "Files"
  InputDevices    "/dev/gpmdata"
  InputDevices    "/dev/input/mice"
  FontPath      "/usr/share/fonts/misc:unscaled"
  FontPath      "/usr/share/fonts/75dpi:unscaled"
  FontPath      "/usr/share/fonts/100dpi:unscaled"
  FontPath      "/usr/share/fonts/Type1"
  FontPath      "/usr/share/fonts/URW"
  FontPath      "/usr/share/fonts/Speedo"
  FontPath      "/usr/share/fonts/cyrillic"
  FontPath      "/usr/share/fonts/truetype"
  FontPath      "/usr/share/fonts/uni:unscaled"
```

```
  FontPath      "/opt/kde3/share/fonts"
  FontPath      "/usr/local/share/fonts"
EndSection
```

La sección **Files** es optativa. Los fabricantes suelen respetar rutas por defecto y bien conocidas. En particular se buscan las fuentes de caracteres de manera automática en /usr/lib/X11/fonts. No obstante, algunas distribuciones modifican a menudo los scripts de ejecución de X para facilitar otras rutas. En las últimas Mandriva, por ejemplo, las fuentes están todas en /usr/share/fonts.

Las versiones recientes de Xorg van a buscar de manera automática las rutas en el directorio /etc/X11/fontpath.d. Las rutas son vínculos simbólicos hacia los directorios correspondientes de /usr/share/fonts y retoman el formato de las entradas **FontPath**.

j. Sección Modules

La sección **Modules** facilita al servidor X una lista de módulos complementarios y optativos que se han de cargar para añadirle nuevas funcionalidades. Se declara un módulo con una línea **Load**.

```
Section "Module"
  Load     "dbe"
  Load     "type1"
  Load     "freetype"
  Load     "extmod"
  Load     "glx"
EndSection
```

A continuación presentamos una lista de módulos habituales. La mayoría son optativos, pero algunos componentes (periféricos de entrada, aplicaciones, tarjetas gráficas) pueden necesitar ciertas extensiones para funcionar. ¿Cómo hacer funcionar OpenOffice.org sin el soporte de las fuentes de caracteres por ejemplo?

- **dbe** (*Double Buffer Extension*) dos búferes de visualización. Uno, principal, al que le corresponde la visualización actual, y otro, secundario, la visualización en segundo plano. Una vez que está listo, se conmutan los dos búferes. Este mecanismo evita que la imagen se cuelgue.
- **extmod**: módulo de extensión del protocolo X, que (casi) todo el mundo utiliza. Al final, se debería incorporar en el mismo interior de X.
- **freetype**: permite utilizar fuentes de caracteres TrueType (ttf).
- **type1**: permite utilizar las fuentes Type1.
- **bitmap**: permite utilizar las fuentes bitmap (innecesario para las últimas versiones).
- **GLcore**: modo básico de añadido de las extensiones OpenGL.
- **glx**: extensiones GLX (extensión a OpenGL).
- **dri** (*Direct Rendering Infrastructure*): OpenGL llama a las funciones de la tarjeta gráfica; se acelera considerablemente la visualización en 3D.
- **i2c**: instalación del bus serie i2c, para comunicarse, entre otros, con el monitor.
- **ddc**: protocolo DDC para los monitores (*Display Data Channel*), que pasa por el bus i2c.
- **int10**: capa de emulación/acceso en tiempo real a la interrupción 10 de la tarjeta gráfica, en particular, para acceder a las funcionalidades VESA de la tarjeta y del monitor.
- **vbe** (*Vesa Bios Extension*): extensiones Vesa para los accesos a algunos modos y resolución de la tarjeta.

k. Sección ServerFlags

La sección **ServerFlags** permite definir las opciones globales del servidor X Window. Las opciones

son múltiples y aquí le presentamos algunas:

- **DontZap**: si es verdadero, desactiva la secuencia Alt-Ctrl-Backspace (que detiene al servidor X).
- **DontVTSwitch**: si es verdadero, desactiva el acceso a las consolas mediante Alt-Ctrl-Fn.
- **DontZoom**: si es verdadero, prohíbe cambiar la resolución con Alt-Ctrl-+/-.
- **AllowMouseOpenFail**: si es verdadero, X se inicia incluso sin ratón.
- **XkbDisable**: si es verdadero, desactiva el teclado.
- **Xinerama**: activa el soporte dual/multi-head.
- **AIGLX**: activa el soporte AIGLX (depende del driver).
- **BlankTime**: duración en minutos por defecto de la actividad del ahorro de pantalla antes de pasar al modo Stand by de ahorro de energía.
- **StandbyTime**: en minutos, duración de la fase Stand by del ahorro de energía DPMS.
- **SuspendTime**: igual, pero para la suspensión.
- **Offtime**: igual, pero para el apagado.

El ejemplo siguiente muestra una sección que permite iniciar X sin ratón (nada impide utilizar luego un ratón USB o inalámbrico), con prohibición de parar el servidor, cambiar de resolución y pasar a consola. Es la configuración por defecto de un nodo de Internet en Linux.

```
Section "ServerFlags"
  Option    "AllowMouseOpenFail" "on"
  Option    "DontZap" "on"
  Option    "DontZoom" "on"
  Option    "DontVTSwitch" "on"
EndSection
```

l. Sección Extensions

Sección optativa, Extensions permite activar o desactivar extensiones de Xorg. Las más conocidas son Damage y Composite: van juntas.

La extensión Damage permite señalar las ventanas que se debe volver a dibujar una parte de su visualización. Si se activan las extensiones compuestas (efecto de transparencia, por ejemplo) se puede volver a dibujar una zona de la pantalla mientras permanece cubierta por una ventana cuyo contenido no ha cambiado pero que puede ser transparente: es la de debajo que ha cambiado. La extensión Damage señala el cambio. Suele ser activada por defecto.

La opción compuesta es la que permite tener los efectos del mismo nombre: sombras, transparencia de las ventanas, alpha-blending, etc. Asociada a las 3D, en particular AIGLX, le permite obtener, con los efectos propuestos por Compiz-Fusion, cubos 3D, efectos de escritorio. No siempre es necesario tener una máquina y tarjeta potentes: algunas tarjetas antiguas funcionan muy bien.

```
Section "Extensions"
  Option     "Composite" "Enable"
EndSection
```

m. xorg.conf.d

Hay un método mejor que poner toda la configuración en un único archivo xorg.conf. La configuración se puede modularizar en varios archivos de configuración: uno para los módulos, otro para las entradas y salidas y así en adelante. Estos archivos siguen la misma sintaxis vista hasta ahora. Pueden tener cualquier nombre, pero deben tener ".conf" como sufijo y estar ubicados en /etc/X11/xorg.conf.d o /usr/lib/X11/xorg.conf.d. El manual de xorg.conf proporciona una lista completa de las rutas posibles para estos archivos.

n. xorg.conf al completo

A continuación, una configuración completa del archivo xorg.conf:

- Tarjeta Nvidia.
- Pantalla LG 19 pulgadas.
- Resolución de 1.280x1.024.
- 16 millones de colores.
- Teclado español de 105 teclas.
- Ratón con 10 botones.
- Extensiones compuestas activadas.

```
Section "ServerLayout"
  Identifier     "Layout[all]"
  Screen         "Screen[0]" 0 0
  InputDevice    "Keyboard[0]" "CoreKeyboard"
  InputDevice    "Mouse[1]" "CorePointer"
EndSection

Section "Files"
  InputDevices   "/dev/gpmdata"
  InputDevices   "/dev/input/mice"
  FontPath     "/usr/share/fonts/misc:unscaled"
  FontPath     "/usr/share/fonts/75dpi:unscaled"
  FontPath     "/usr/share/fonts/100dpi:unscaled"
  FontPath     "/usr/share/fonts/Type1"
  FontPath     "/usr/share/fonts/URW"
  FontPath     "/usr/share/fonts/Speedo"
  FontPath     "/usr/share/fonts/cyrillic"
  FontPath     "/usr/share/fonts/truetype"
  FontPath     "/usr/share/fonts/uni:unscaled"
  FontPath     "/opt/kde3/share/fonts"
  FontPath     "/usr/local/share/fonts"
EndSection

Section "Module"
  Load     "dbe"
  Load     "type1"
  Load     "freetype"
  Load     "extmod"
  Load     "glx"
EndSection

Section "ServerFlags"
  Option    "AllowMouseOpenFail" "on"
  Option    "DontZap" "on"
  Option    "DontZoom" "on"
  Option    "DontVTSwitch" "on"
EndSection

Section "InputDevice"
  Identifier   "Keyboard[0]"
  Driver       "kbd"
  Option       "Protocol" "Standard"
  Option       "XkbLayout" "es"
  Option       "XkbModel" "pc105"
  Option       "XkbOptions" "caps:shiftlock"
  Option       "XkbRules" "xfree86"
EndSection

Section "InputDevice"
```

```
  Identifier   "Mouse[1]"
  Driver       "evdev"
  Option       "Buttons" "10"
  Option       "InputFashion" "Mouse"
  Option       "Name" "Logitech USB R*"
  Option       "Protocol" "event"
  Option       "SendCoreEvents" "on"
  Option       "Vendor" "Sysp"
  Option       "ZAxisMapping" "4 5"
EndSection

Section "Modes"
  Identifier   "Modes[0]"
  ModeLine     "1280x1024" 102.6 1280 1312 1472 1632 1024 1028
1032 1048 -hsync -vsync
EndSection

Section "Monitor"
  Identifier    "Monitor[0]"
  VendorName    "--> VESA"
  ModelName     "1280X1024@60HZ"
  UseModes      "Modes[0]"
  DisplaySize   340 270
  HorizSync     31.0 - 64.0
  VertRefresh   50.0 - 60.0
  Option        "DPMS"
EndSection

Section "Device"
  Identifier   "Device[0]"
  Driver       "nvidia"
  VendorName   "NVidia"
  BoardName    "GeForce 8600 GT"
  Option       "RenderAccel" "True"
  Option       "AddARGBGLXVisuals" "True"
EndSection

Section "Screen"
  Identifier   "Screen[0]"
  Device       "Device[0]"
  Monitor      "Monitor[0]"
  DefaultDepth  24
  SubSection   "Display"
    Depth       15
    Modes      "1280x1024"
  EndSubSection
  SubSection   "Display"
    Depth       16
    Modes      "1280x1024"
  EndSubSection
  SubSection   "Display"
    Depth       24
    Modes      "1280x1024"
   EndSubSection
   SubSection  "Display"
    Depth       8
    Modes      "1280x1024"
   EndSubSection
EndSection

Section "Extensions"
    Option    "Composite" "Enable"
EndSection
```

5. Probar e iniciar X

a. Comprobar la configuración

Xorg 7.2 o anterior

Una vez terminada su configuración de X, es el momento de probar el servidor e iniciarlo. Si tiene una versión de Xorg 7.2 o anterior, siga este proceso. Para ello, obviamente, no debe estar ya en modo gráfico. Pase a modo texto (init 2 o 3 según su distribución) y teclee:

```
$ X -probeonly

X Window System Version 7.2.0
Release Date : TRue Jan 22 17:08:26 UTC 2008-05-31
X Protocol Version 11, Revision 0, Release 7.2
Build Operating System: openSUSE SUSE LINUX
Current Operating System : Linux opensuse 2.6.22.17-0.1-default #1
SMP 2008/02/10 20:01:04 UTC i686
Build Date: 22 Junuary 2008
    Before reporting problems, check http://wiki.x.org
    To make sure that you have the lastest version.
Module Loader present
Markers: (--) probed, (**) from config file, (==) default setting,
    (++) from command line, (!!) notice, (II) informational,
    (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: ”/var/log/Xorg.0.log”, Time: Sat May 31 10:16:06 2008
(==) Using config file: ”/etc/X11/xorg.conf”
(II) Module already build-in
```

No se ha detectado ningún error. De haber habido uno, le saldría esto:

```
(EE) No drivers available.

Fatal server error:
no screens found
```

En ese caso:

- O el archivo **/etc/X11/xorg.conf** contiene un error de sintaxis.
- O se ha configurado mal un periférico: pantalla ausente, ninguna resolución soportada, driver incorrecto, etc.

El análisis del registro de sucesos del servidor X es de gran ayuda.

Xorg 7.3 o posterior

Si utiliza una versión 7.3 o posterior de Xorg, puede intentar realizar la autoconfiguración iniciando el comando siguiente:

```
# Xorg --configure
```

Este comando producirá un archivo xorg.conf.new en el directorio de root. Puede modificarlo si así lo desea. Pruebe a continuación este archivo indicando específicamente su uso:

```
# Xorg --config xorg.conf.new
```

Si el servidor X se inicia (puede finalizarlo con [Ctrl][Alt][BackSpace]), renómbrelo y copie este archivo a **/etc/X11/xorg.conf**. Ésta será la configuración por defecto.

b. El registro

Las entradas en el registro de eventos (trazas) se encuentran en el

archivo**/var/log/Xorg.0.log**. Los valores numéricos pueden variar: como un equipo puede arrancar varios servidores X (un par de pantallas por ejemplo) el archivo podrá entonces ser Xorg.1.log u otro. Las trazas son, a menudo, muy largas: con la configuración anterior, el registro contiene 600 trazas y se pueden añadir otras durante el tiempo de funcionamiento. Las entradas en el registro contienen todos los detalles de la carga y de la configuración de X Window.

En caso de error, las últimas líneas suelen ser elocuentes e indican dónde se encuentra el problema. Si X funciona pero no reacciona como debería, busque toda la información en las trazas. Ejemplos interesantes de trazas los proporcionan la tarjeta gráfica y el teclado. Puede ver en negrita qué ha detectado Xorg: una tarjeta Nvidia 8600GT, su versión de BIOS, el bus PCI express, la cantidad de memoria y la marca y el modelo del monitor.

```
[    19.835]
X.Org X Server 1.10.1
Release Date: 2011-04-15
[    19.835] X Protocol Version 11, Revision 0
[    19.835] Build Operating System: Linux 2.6.24-29-server x86_64 Ubuntu
[    19.835] Current Operating System: Linux slyserver 2.6.38-11-generic #48-Ubuntu
SMP Fri Jul 29 19:02:55 UTC 2011 x86_64
[    19.835] Kernel command line: BOOT_IMAGE=/vmlinuz-2.6.38-11-generic
root=UUID=ecb82159-84eb-494e-aa1b-41c8f9e9df63 ro quiet splash nomodeset
video=uvesafb:mode_option=1280x1024-24,mtrr=3,scroll=ywrap security=selinux selinux=1
[    19.835] Build Date: 21 May 2011  11:48:41AM
[    19.835] xorg-server 2:1.10.1-1ubuntu1.1
(For technical support please see http://www.ubuntu.com/support)
[    19.835] Current version of pixman: 0.20.2
[    19.835]    Before reporting problems, check http://wiki.x.org
        to make sure that you have the latest version.
[    19.835] Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
[    19.835] (==) Log file: "/var/log/Xorg.0.log", Time: Tue Aug 23 18:01:40 2011
[    19.836] (==) Using config file: "/etc/X11/xorg.conf"
[    19.836] (==) Using system config directory "/usr/share/X11/xorg.conf.d"
[    19.836] (==) No Layout section.  Using the first Screen section.
[    19.836] (**) |-->Screen "Default Screen" (0)
...
[    19.900] (**) NVIDIA(0): Depth 24, (--) framebuffer bpp 32
[    19.900] (==) NVIDIA(0): RGB weight 888
[    19.900] (==) NVIDIA(0): Default visual is TrueColor
[    19.900] (==) NVIDIA(0): Using gamma correction (1.0, 1.0, 1.0)
[    19.900] (**) NVIDIA(0): Option "NoLogo" "True"
[    20.687] (II) NVIDIA(GPU-0): Display (LG Electronics L1915S (CRT-0)) does not
support
[    20.687] (II) NVIDIA(GPU-0):     NVIDIA 3D Vision stereo.
[    20.690] (II) NVIDIA(0): NVIDIA GPU GeForce 8600 GT (G84) at PCI:1:0:0 (GPU-0)
[    20.690] (--) NVIDIA(0): Memory: 262144 kBytes
[    20.690] (--) NVIDIA(0): VideoBIOS: 60.84.35.00.11
[    20.690] (II) NVIDIA(0): Detected PCI Express Link width: 16X
[    20.690] (--) NVIDIA(0): Interlaced video modes are supported on this GPU
[    20.690] (--) NVIDIA(0): Connected display device(s) on GeForce 8600 GT at
PCI:1:0:0
...
[    21.051] (II) Using input driver 'evdev' for 'USB Video Camera'
[    21.051] (II) Loading /usr/lib/xorg/modules/input/evdev_drv.so
[    21.051] (**) USB Video Camera: always reports core events
[    21.051] (**) USB Video Camera: Device: "/dev/input/event5"
[    21.090] (--) USB Video Camera: Found keys
[    21.090] (II) USB Video Camera: Configuring as keyboard
[    21.090] (**) Option "config_info"
"udev:/sys/devices/pci0000:00/0000:00:1a.7/usb1/1-5/1-5:1.0/input/input5/event5"
[    21.090] (II) XINPUT: Adding extended input device "USB Video Camera"
(type: KEYBOARD)
[    21.090] (**) Option "xkb_rules" "evdev"
```

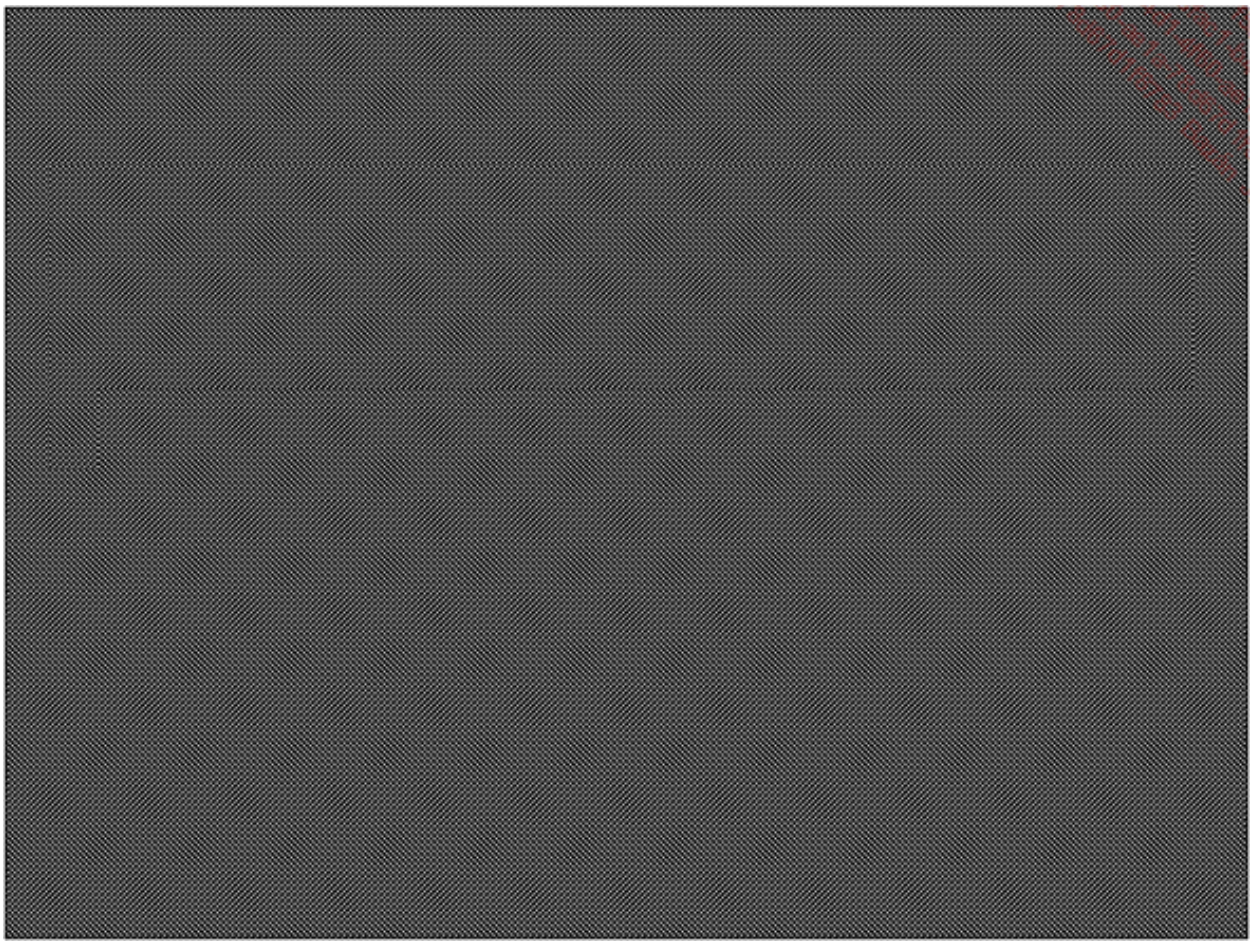
```
[    21.090] (**) Option "xkb_model" "pc105"
[    21.090] (**) Option "xkb_layout" "fr"
[    21.090] (**) Option "xkb_variant" "oss"
[    21.091] (II) config/udev: Adding input device HDA Intel Headphone
(/dev/input/event6)
...
```

c. Probar el servidor

Pruebe el servidor X iniciándolo con su nombre:

```
$ X
```

Si X funciona, debe tener una pantalla en gris, en realidad una sucesión de puntos negros y blancos, y una cruz en medio que es el cursor del ratón. Mueva el ratón. Si funciona, es que se ha configurado bien su servidor.



Una pantalla gris: X funciona

Detenga el servidor X con [Ctrl][Alt][Retroceso], porque sin ningún gestor de ventanas no hay nada que hacer.

El Display Manager

1. Principio

El Display Manager, o gestor de visualización, es un elemento de X Window que se encarga de la conexión de usuarios, locales o remotos, de su autenticación, y de la carga de componentes en su entorno de trabajo al abrir sesión. Gestiona un conjunto de visualizaciones X que pueden ser locales o remotas.

Para las conexiones remotas, el Display Manager se apoya en el protocolo estándar XDMCP (*X Display Manager Control Protocol*).

El Display Manager representa más o menos el equivalente gráfico de los servicios propuestos por init, getty y login: pide identificadores y contraseñas, autentica a las personas y abre una sesión.

El gestor por defecto se llama XDM: X Display Manager. Su estilo gráfico no resulta muy atractivo, pero es ligero y funciona con todos los servidores X.

Se puede ver una sesión X como una sesión de consola: es la duración del proceso de un usuario después de la conexión. En una consola es el shell; en X, es una "session manager", en general, el gestor de ventanas (o un proceso del entorno ofimático) o un terminal gráfico. Cuando este proceso está cerrado (desconexión del terminal, salida del entorno ofimático, etc.) la sesión finaliza y el sistema vuelve a mostrar la caja de conexión de XDM (o cualquier otro Display Manager).

Es posible abrir varias sesiones X desde un solo Display Manager, incluso en una misma máquina.

Linux dispone de varios Display Manager, pero se utilizan principalmente tres:

- XDM: versión básica.
- GDM: versión propuesta por GNOME.
- KDM: versión ofrecida por KDE.

Las versiones GNOME y KDE han evolucionado: ofrecen lo mismo que XDM con funcionalidades adicionales:

- lista de los usuarios;
- iconos (avatares) asociados;
- elección de una sesión gráfica particular (Gnome, fvwm, KDE, etc.);
- posibilidad de autoconexión;
- temas gráficos atractivos;
- lista de los servidores X remotos (XDMCP);
- pasar de un usuario a otro;
- etc.

2. XDM

a. Configuración general

Los archivos de configuración de XDM se encuentran en el directorio **/etc/X11/xdm**. El primer archivo cargado es xdm-config. Todas sus líneas son del tipo:

```
DisplayManager.recurso: valor.
```

Cada línea representa un recurso de XDM. La certificación LPI1 no exige saber de memoria la configuración de XDM, sólo algunos elementos básicos, como xdm-config, que carga otros archivos. En el ejemplo siguiente, las líneas en negrita muestran otros scripts de configuración llamados, o interpretados directamente por xdm, o ejecutados antes o después de la conexión:

```
!! xdm-config: Configuration of the xdm
!
DisplayManager.errorLogFile:        /var/log/xdm.errors
DisplayManager.pidFile:             /var/run/xdm.pid
DisplayManager.authDir:             /var/lib/xdm
DisplayManager.keyFile:             /etc/X11/xdm/xdm-keys
DisplayManager.servers:             /etc/X11/xdm/Xservers
DisplayManager.accessFile:          /etc/X11/xdm/Xaccess
DisplayManager.willing:             su nobody -c /etc/X11/xdm/Xwilling
!
! ATTENTION: `authName' should be in general MIT-MAGIC-COOKIE-1
! For XDM-AUTHENTICATION-1 which is default for xterminals see
! manual page of xdm and the manual coming with the xterminal.
!
DisplayManager.*.authName:          MIT-MAGIC-COOKIE-1
DisplayManager.*.authComplain:      false
!
! All displays should use authorization, but we cannot be sure
! X terminals will be configured that way, so by default
! use authorization only for local displays :0, :1, etc.
!
DisplayManager._0.authorize:        true
DisplayManager._1.authorize:        true
DisplayManager._93.authorize:       true
!
! The scripts handling the setup, the startup, the session its self,
! and the reset of an X session.
!
DisplayManager.*.setup:              /etc/X11/xdm/Xsetup
DisplayManager.*.chooser:            /etc/X11/xdm/RunChooser
DisplayManager.*.startup:            /etc/X11/xdm/Xstartup
DisplayManager.*.session:            /etc/X11/xdm/Xsession
DisplayManager.*.reset:              /etc/X11/xdm/Xreset
!
DisplayManager._0.terminateServer:   true
DisplayManager._93.terminateServer:  true
!
DisplayManager*resources:            /etc/X11/xdm/Xresources
DisplayManager.*.terminateServer:    false
!
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
!
!DisplayManager.requestPort:         0
```

b. Setup: Xsetup

El archivo **/etc/X11/xdm/Xsetup** o cualquier otro asociado a la entrada**DisplayManager.*.setup** es invocado antes incluso de que el sistema muestre la ventana de login. Es en este archivo donde puede determinar la presentación gráfica de XDM o cómo visualizar nuevas ventanas (registro del sistema, relojes, etc.) El contenido por defecto de este archivo depende del editor de la distribución, pero en general todos ellos lo usan para:

- modificar el fondo de pantalla con el comando xsetroot;
- modificar los colores, con xrdb;
- cambiar con los recursos X la geometría (las posiciones y tamaños) de los cuadros de diálogo;

- visualizar una consola con xconsole;
- activar el teclado numérico con numlockx;
- modificar el teclado con xmodmap;
- etc.

Aquí tiene un ejemplo depurado procedente una vez más de openSUSE. Las líneas en negrita muestran las acciones para xdm.

```
#
# Handle background:
#   First kdm/gdm choise, then xdm/user choise and
#   if no choise is given use the system defaults.
# Pantalla de selección
if test "$kdm" = "yes" -o "$gdm" = "yes" ; then
    : # $xsetroot -solid '#738dc6'
elif test -s ${background}.gz -a -x $xpmroot ; then
    $xpmroot ${background}.gz
elif test -s ${background} -a -x $xpmroot ; then
    $xpmroot $background
elif test -x $backprg ; then
    $backprg
else
    $xsetroot -gray
fi

#
# Enable Numlock if set
# Activación de la extension del teclado numérico
if test -r /var/run/numlock-on && type -p numlockx > /dev/null ;
then
    numlockx on
fi

#
# Xresources
# Modificación de colores y geometría de la pantalla
if test "$kdm" != "yes" -a "$gdm" != "yes" ; then
    $xrdb -override -retain <<-EOF
      #ifdef COLOR
      *Foreground: black
      *Background: #cdd2b4
      #endif
      #if (WIDTH < 320)
      XConsole*geometry:   125x80+0-0
      #elif (WIDTH < 400)
      XConsole*geometry:   130x85+0-0
      #elif (WIDTH < 640)
      XConsole*geometry:   150x90+0-0
      #elif (WIDTH < 800)
      XConsole*geometry:   240x95+0-0
      #elif (WIDTH < 1024)
      XConsole*geometry:   300x100+0-0
      #elif (WIDTH < 1152)
      XConsole*geometry:   384x110+0-0
      #elif (WIDTH < 1280)
      XConsole*geometry:   432x120+0-0
      #else
      XConsole*geometry:   480x130+0-0
      #endif
      EOF
fi
```

```
if test "$kdm" = "yes" -o "$gdm" = "yes" ; then
    $xrdb -override -retain ${ETCDIR}/Xresourcesf
i
#
# The geometry of xconsole is set in the Xresource file.
# Inicio de una consola
(
    exec setsid $xconsole -notify -nostdin -verbose -exitOnFail
) & echo $! > /var/run/xconsole.pid
```

c. Chooser: RunChooser

El archivo **/etc/X11/xdm/RunChooser** o cualquier otro asociado con la entrada**DisplayManager.*.chooser** permite visualizar el cuadro de diálogo de bienvenida de los servidores X remotos para conectarse a ellos. Se parece al archivo Xsetup, pero se limita a la configuración y al inicio del programa **chooser** (/usr/X11R6/bin/chooser, /usr/lib/X11/chooser o cualquier otro programa que tenga una función similar). Este programa tiene interés únicamente si el servidor X remoto acepta las conexiones mediante XDMCP.

d. Startup: Xstartup

Tras la autenticación del usuario, el sistema busca el archivo **/etc/X11/xdm/Xstartup** o cualquier otro asociado con la entrada **DisplayManager.*.startup** y lo ejecuta. Se ejecuta con privilegios de root. No debe confundirse con el archivo de sesión, del cual hablaremos más adelante. Sirve, entre otras cosas, para:

- borrar la pantalla;
- escribir la información de conexión en los archivos /var/log adecuados;
- comprobar si la conexión es local o remota;
- comprobar si se autoriza al usuario a conectarse;
- etc.

e. Sesión: Xsession

Tras la autenticación del usuario y ya con el conjunto de privilegios de éste, se inicia el archivo**/etc/X11/xdm/Xsession** o cualquier otro asociado con la entrada**DisplayManager.*.session**. El sistema comprobará la instalación del servidor X; si no está instalado, ejecutará un comando **Xterm** y lo asociará al proceso de sesión del usuario. De hecho, si una sesión de usuario no se pudiera iniciar con normalidad, este archivo contiene lo necesario para iniciar una consola. Después de una configuración básica, Xsession intenta iniciar en este orden los archivos siguientes:

- el archivo $HOME/.xsession, propio del usuario;
- si no lo encuentra, el archivo $HOME/.xinitrc del usuario;
- si no lo encuentra, el archivo /etc/X11/xdm/sys.xsession;
- si no lo encuentra, el archivo /etc/X11/xinit/xinitrc.

Mediante una bandera (flag) se puede forzar un arranque por defecto e ir directamente a la carga de un Window Manager (KDE o GNOME) sin pasar por los archivos anteriores. Los entornos gráficos más evolucionados tienen automatizada su propia configuración.

```
#
# Forced X session type if the user asked for
# an other session environment.
#
if test "$forced" = "yes" ; then
```

```
    unset WINDOW_MANAGER STARTUP
     test -x $syssess && exec_login "$syssess"
     exec_login "/bin/bash $syssess"
fi

# User login X session
# If the user doesn't have their own xsession, then run
# system xsession or xinitrc script if they exist

if test -f $session ; then
   test -x $session && exec_login "$session"
   exec_login "/bin/bash $session"
elif test -f $xinitrc ; then
   test -x $xinitrc && exec_login "$xinitrc"
   exec_login "/bin/bash $xinitrc"
elif test -f $syssess; then
   test -x $syssess && exec_login "$syssess"
   exec_login "/bin/bash $syssess"
elif test -f $sysinit ; then
   test -x $sysinit && exec_login "$sysinit"
   exec_login "/bin/bash $sysinit"
elif test -n "$WINDOWMANAGER" ; then
   unset WINDOW_MANAGER STARTUP
   exec_login "$WINDOWMANAGER"
fi
```

f. Reset: Xreset

Cuando un usuario cierra la sesión, el sistema inicia el archivo **/etc/X11/xdm/Xreset** o cualquier otro asociado con la entrada **DisplayManager.*.reset**. En el registro del sistema encontrará entradas procedentes de este archivo.

g. Resources: Xresources

El archivo **/etc/X11/xdm/Xresources** o cualquier otro asociado con la entrada**DisplayManager*resources** con xdm contiene las definiciones de los recursos visuales utilizados por xdm, como las fuentes de caracteres, los mensajes de bienvenida o error, los colores, la adaptación de la visualización en función del número de colores y el tamaño de la pantalla, etc.

En particular, en este archivo puede modificar el mensaje de bienvenida o de error:

```
xlogin*titleMessage:   Xlogin
xlogin*greetColor:     darkgray
xlogin*promptColor:    darkgray
xlogin*failColor:      red
xlogin*greeting:       Bienvenido/a a CLIENTHOST
xlogin*fail:           -- Conexión denegada -
```

Con una configuración como la anterior, el sistema recibirá al usuario con el mensaje "Bienvenido/a a CLIENTHOST", que vemos en gris oscuro. Si no se autentica, se visualiza en rojo el mensaje "-- Conexión denegada".

h. Servers: Xservers

El archivo **/etc/X11/xdm/Xservers** o cualquier otro archivo asociado con la entrada **DisplayManager.servers** proporciona la lista de las especificaciones de los servidores locales de X o, más bien, de los que no necesitan conexión mediante XDMCP. La línea siguiente indica que el primer servidor :0 es local, su binario es /usr/bin/X con opción -br vt7 (ocupa el terminal virtual vt7).

```
:0 local /usr/bin/X -br vt7
```

En esta línea, puede, si lo desea, modificar el número de colores del servidor X durante el inicio de xdm pasándole los argumentos correctos. Para cambiar a 256 colores:

```
:0 local /usr/bin/X -br vt7 -depth 8
```

i. AccessFile: Xaccess y XDMCP

El archivo **/etc/X11/xdm/Xaccess** o cualquier otro archivo asociado con la entrada**DisplayManager.accessFile** proporciona la lista de los anfitriones autorizados a conectarse mediante XDMCP a su servidor X. Para autorizar conexiones, tendrá que modificar dos archivos:

- En **/etc/X11/xdm/xdm-config**, quite el comentario a la línea '!DisplayManager.requestPort', de tal manera que quede como a continuación:

```
$ grep request /etc/X11/xdm/xdm-config
...
DisplayManager.requestPort: 0
```

- En **/etc/X11/xdm/Xaccess**, quite el comentario a la línea que empieza con un asterisco:

```
* CHOOSER BROADCAST
```

O aún más fácil:

```
*
```

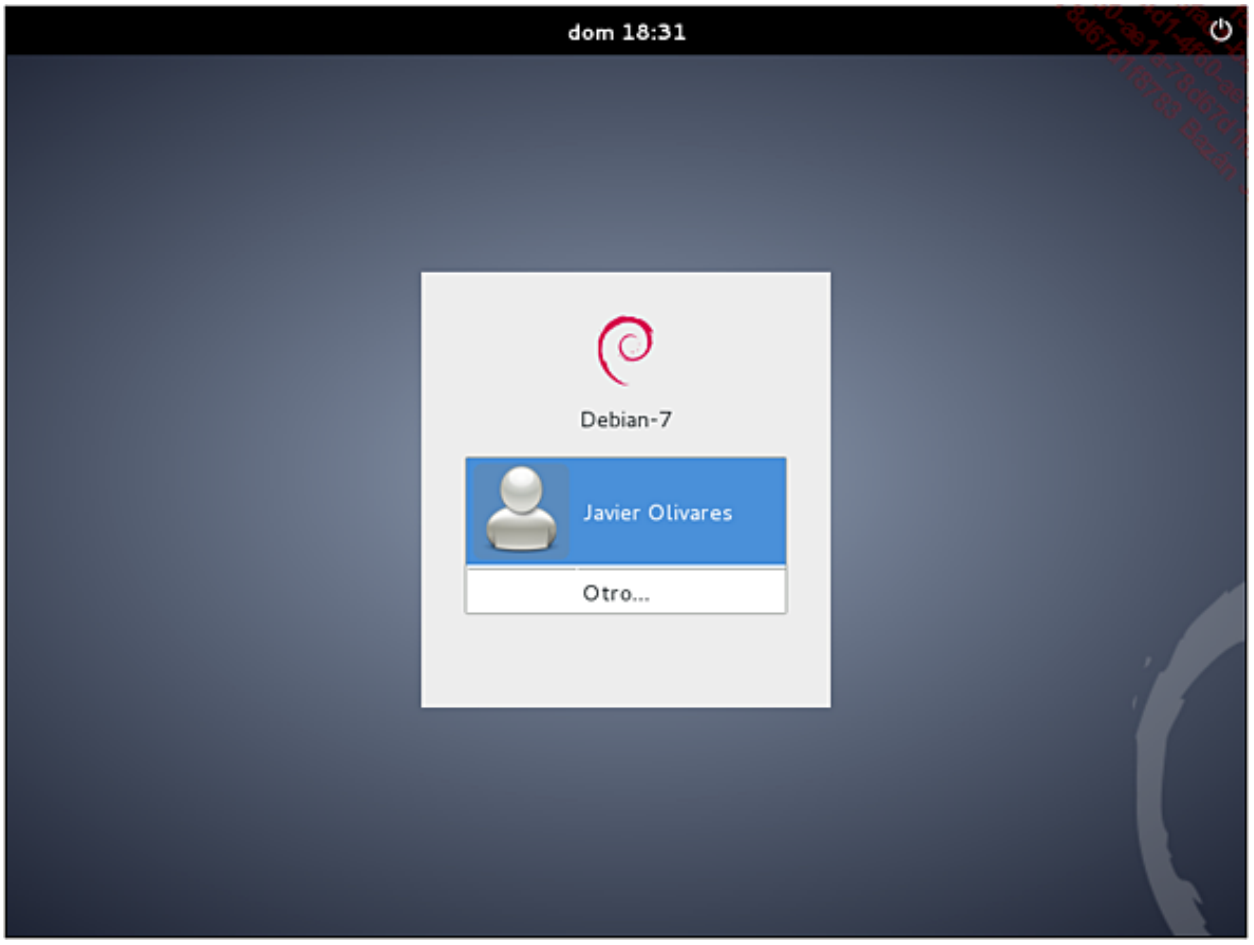
Esto es todo. Todos los anfitriones podrán conectarse a su máquina.

3. gdm y kdm

gdm y **kdm** son, respectivamente, los gestores de visualización de GNOME y KDE.

¿Por qué hemos dedicado tanto tiempo a xdm cuando gdm y kdm son mucho más bonitos, prácticos y eficaces (aquí puede añadir tantos piropos superlativos como quiera...)? Porque, excepto por supuesto para todos los recursos gráficos propios, gdm y kdm emplean (aunque no sea una obligación) los mismos archivos que xdm, entre los cuales figuran:

- Xsetup
- Xstartup
- Xsession
- Xreset
- Xresources
- Xaccess
- etc.



GDM arrancado desde Debian

La configuración avanzada de gdm y kdm se salta los archivos comunes de XDM (que configura de forma automática), pero permite configurar manualmente aquéllos propios del gestor, ya sea mediante archivos de texto o, si las modificaciones son sencillas, pasando por la interfaz gráfica asociada.

- La configuración de KDM se coloca en **/etc/X11/kdm/kdmrc** o en **<prefix-kde>/share/config/kdmrc**, por ejemplo en openSUSE el archivo se coloca en**/opt/kde3/share/config/kdm/kdmrc**. A partir de KDE4 el archivo se ubica en**/etc/KDE4/kdm/kdmrc** o en **/etc/kde/kdem/kdmrc**.
- La configuración de GDM se coloca en **/etc/X11/gdm/gdm.conf**, o**/etc/gdm/gdm.conf** o también en **/usr/share/gdm/gdm.conf**.

Aunque gdm y kdm utilizan por defecto las opciones de los archivos de XDM, puede modificar la configuración de manera que se abstraiga totalmente de ellas. Por ejemplo, para kdm, puede activar XDMCP así:

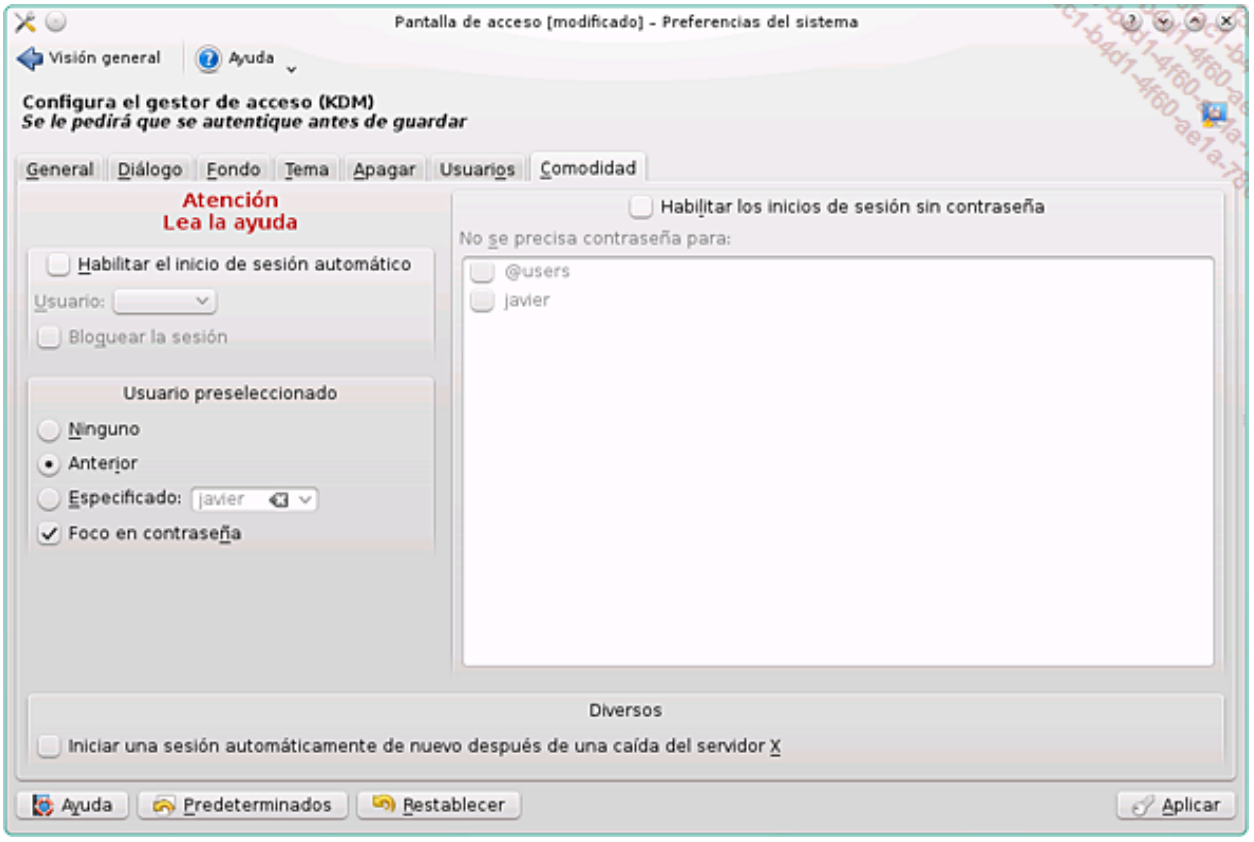
```
[Xdmcp]
Enable=true
Port=177
KeyFile=/etc/kde4/kdm/kdmkeys
Xaccess=/etc/kde4/kdm/Xaccess
```

Otra (falsa) buena idea es autorizar la exportación de las ventanas X hacia su servidor X (visualización remota) mediante el comando **xhost+** suprimiendo el parámetro **nolisten** de la configuración. Comente la línea siguiente:

```
ServerArgsLocal=-nolisten tcp
```

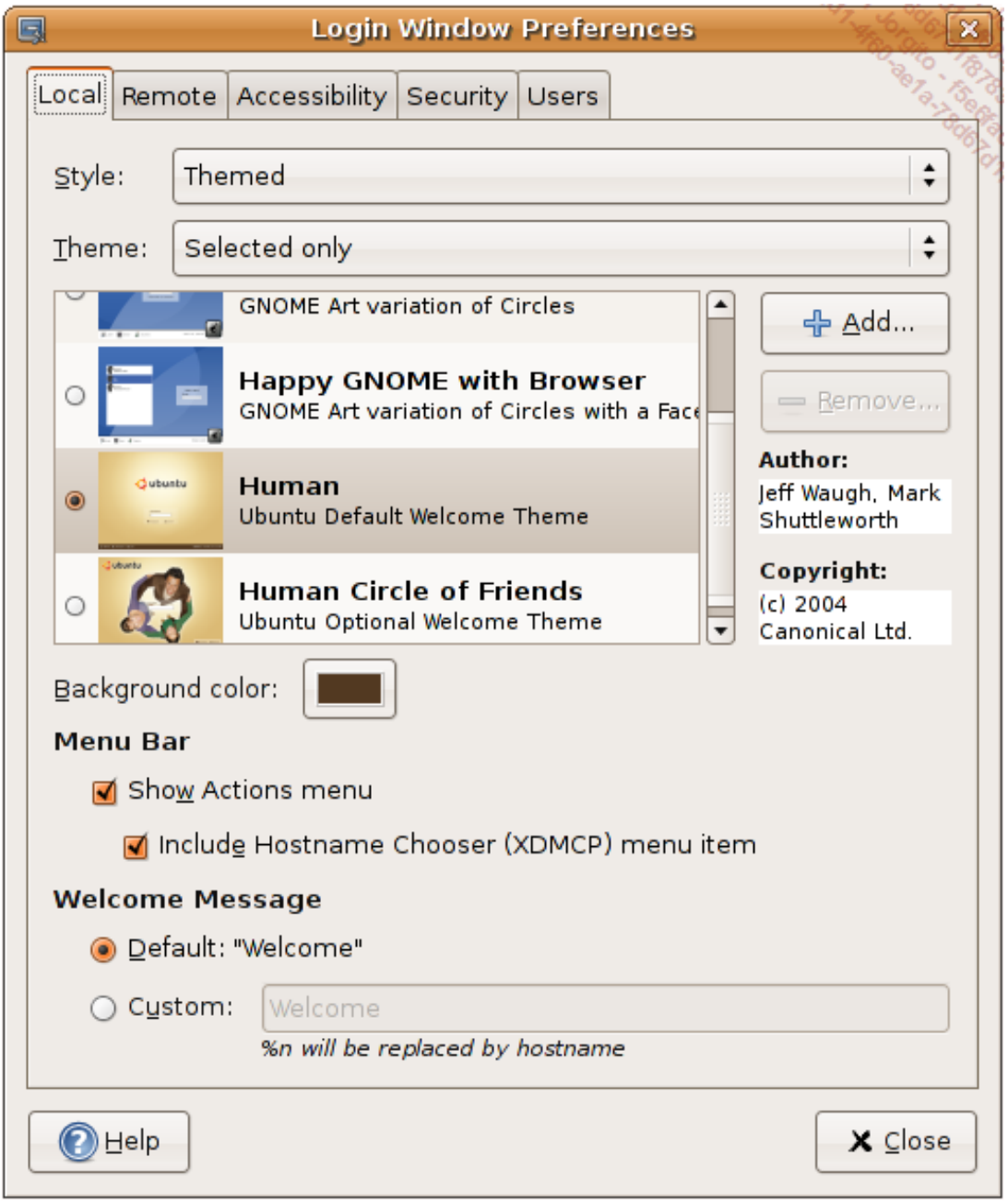
KDM arrancado desde Kubuntu 8.04

Para el resto, use la herramienta de configuración gráfica. Inicie el **Centro de configuración de KDE**, y despliegue la entrada **Administración del sistema**. Haga clic en **Gestor de conexión**. Como usuario genérico no podrá hacer nada: haga clic en el botón **Modo superusuario**, abajo.



A partir de ese punto, puede configurar kdm completamente.

En gdm ejecute, desde la línea de comandos, **gdmsetup**, que inicia la ventana de configuraciones. El ejemplo siguiente procede de una instalación Ubuntu 6.06.



Personalización de GDM

gdmsetup no está disponible en las últimas versiones. Puede descargar e instalar la herramienta gdm3setup disponible desde el enlace: https://github.com/Nano77/gdm3setup

4. xdm, gdm o kdm al boot

a. inittab

Para iniciar X y el gestor de ventanas durante el proceso de arranque del sistema, los administradores cuentan con dos opciones en la mayoría de las distribuciones:

- editar **`/etc/inittab`**;

- o iniciar xdm como servicio.

En el primer caso, abra el archivo **/etc/inittab** y busque una línea que se parezca a una de las líneas siguientes. Encontrará tanto las líneas once como `respawn`, siendo preferible el segundo caso:

```
xdm:5:once:/usr/X11R6/bin/xdm
xdm:5:respawn:/usr/X11R6/bin/xdm
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Si el archivo **inittab** contiene una línea equivalente a la del ejemplo pero sigue accediendo al sistema en modo consola, puede ser debido a que:

- X está mal configurado, y por lo tanto después de algunos intentos devuelve el control a la consola. En ese caso, la pantalla debería parpadear varias veces en este momento;
- Está en el nivel de ejecución (runlevel) incorrecto.

En este último caso, compruebe y modifique la línea siguiente para iniciar en nivel 5 (para las distribuciones rpm) por defecto:

```
id:5:initdefault:
```

y pase como root al nivel de ejecución 5:

```
# telinit 5
```

b. Servicio System V

Alternativamente se puede iniciar `xdm` como servicio. Verifique la siguiente línea en **/etc/init.d**:

```
$ ls /etc/init.d/xdm
/etc/init.d/xdm
```

Verifique la configuración del nivel asociado mediante **rcupdate.d** o **chkconfig**, o manualmente:

```
# chkconfig --list xdm
xdm     0:off 1:off 2:off 3:off 4:off 5:on 6:off
# ls -l /etc/rc.d/rc?.d/S*xdm
lrwxrwxrwx 1 root root 6 may 14 12:22 /etc/rc.d/rc5.d/S10xdm ->
../xdm
```

c. Destino systemd

Con system, se trata de cambiar el destino por defecto. En Fedora apunta en modo texto a **multi-user.target**:

```
# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37  8 jul. 22:36 /etc/systemd/system/default.target ->
/lib/systemd/system/multi-user.target
```

El destino para un arranque en modo gráfico es el siguiente:

```
# ls /lib/systemd/system/graph*.target
/lib/systemd/system/graphical.target
```

Remplace el vínculo simbólico previo para hacer que apunte al destino **graphical.target**:

```
# ln -sf /lib/systemd/system/graphical.target /etc/systemd/system/default.target
```

O simplemente:

```
# systemctl set-default -f graphical.target
```

Para restablecer el modo texto multi usuario, vuelva al destino inicial:

```
# systemctl set-default -f multi-user.target
```

Para cambiar el destino actual, debemos volver a arrancar la sesión gráfica. Para ello emplee isolate:

```
# systemctl isolate graphical.target
```

d. Service upstart

Upstart se encarga de iniciar gdm o lightdm según el gestor de sesión seleccionado por defecto. Hay que ver el contenido de los archivos **/etc/init/gdm.conf** y**/etc/init/lightdm.conf** para introducirlos.

Podemos ver por ejemplo la opción del gestor por defecto que está especificada en**/etc/X11/default-display-manager**:

```
# cat /etc/X11/default-display-manager
/usr/sbin/lightdm
```

Cada servicio verificará si el contenido es correcto. En caso contrario no arrancará. Dicho de otra forma, para desactivar el arranque gráfico, basta con eliminar el archivo, o escribir en él cualquier cosa.

La otra solución consiste en renombrar los archivos, en .bak por ejemplo. Estos no serán reconocidos en adelante por upstart.

La última solución consiste en impedir el arranque de nivel de ejecución 2 y comentar las líneas start con:

```
stop on runlevel [0126]
```

e. /etc/sysconfig

En varias distribuciones, se colocan la elección y los ajustes por defecto de varios gestores de ventanas en archivos de configuración dentro de la carpeta **/etc/sysconfig**. En Mandriva, por ejemplo, se trata del archivo **/etc/sysconfig/desktop**.

```
$ cat /etc/sysconfig/desktop
DISPLAYMANAGER=kdm
```

Con openSUSE, los ajustes son más precisos (en particular para la gestión de la resolución y de los temas) y el archivo es **/etc/sysconfig/displaymanager**. Particularmente encontrará ajustes que sustituyen a los de xdm relativos a los mensajes que se van a visualizar, los accesos XDMCP, etc.

```
$ cat /etc/sysconfig/displaymanager | grep -v ^#
DISPLAYMANAGER_XSERVER=Xorg
DISPLAYMANAGER_XGL_OPTS="-accel glx:pbuffer -accel xv:pbuffer"
DISPLAYMANAGER="kdm"
```

```
DISPLAYMANAGER_REMOTE_ACCESS="no"
DISPLAYMANAGER_ROOT_LOGIN_REMOTE="no"
DISPLAYMANAGER_STARTS_XSERVER="yes"
DISPLAYMANAGER_XSERVER_TCP_PORT_6000_OPEN="no"
DISPLAYMANAGER_AUTOLOGIN=""
DISPLAYMANAGER_PASSWORD_LESS_LOGIN="no"
DISPLAYMANAGER_AD_INTEGRATION="no"
DISPLAYMANAGER_SHUTDOWN="auto"
DISPLAYMANAGER_RANDR_MODE_VGA="auto"
DISPLAYMANAGER_RANDR_MODE_auto="1024x768_60 64.11 1024 1080 1184
1344 768 769 772 795 -HSync +Vsync"
KDM_USERS=""
KDM_GREETSTRING=""
DISPLAYMANAGER_KDM_THEME="SUSE"
```

f. Ubuntu y Debian

Ubuntu y Debian disponen de servicios gdm (o gdm3) y kdm en **/etc/init.d**, que se activan generalmente por defecto. Para modificar el gestor de sesiones por defecto, edite el archivo**/etc/X11/default-display-manager** e indique en la única línea de este archivo la ruta del ejecutable asociado:

```
# cat /etc/X11/default-display-manager
/usr/bin/kdm
```

Window Manager y entorno personal

1. Mediante el Display Manager

El tipo de entorno gráfico cargado por la sesión X se suele determinar desde gdm o kdm (no es habitual utilizar xdm) y el menú asociado. Éstos recuerdan la sesión anterior como sesión por defecto gracias a la presencia del archivo **.dmrc** en su directorio personal:

```
$ pwd
/home/seb
$ cat .dmrc
~$ cat .dmrc

[Desktop]
Session=kde-plasma
Language=es_ES.utf8
Layout=es
```

Tanto si utiliza como gestor gdm o como si emplea kdm, notará que ambos comparten la lista de las sesiones posibles. Es una de las ventajas de Freedesktop: los archivos de descripción de las sesiones están ubicados en **/usr/share/xsessions** (la descripción del archivo kdmrc configura otras ubicaciones alternativas).

```
$ pwd
/usr/share/xsessions
$ ls -ltotal 20
-rw-r--r-- 1 root root 7262 oct 4 2007 fvwm2.desktop
-rw-r--r-- 1 root root 6796 may 28 13:44 kde.desktop
-rw-r--r-- 1 root root 3305 nov 28 2007 twm.desktop
```

Cada archivo **desktop** representa una sesión posible. Las descripciones y los comandos de inicio están presentes en cada uno de los archivos, y en particular mediante las líneas Exec y Name:

```
$ grep -E "Exec|Name=" *.desktop
fvwm2.desktop:Exec=fvwm
fvwm2.desktop:TryExec=fvwm
fvwm2.desktop:Name=FVWM
kde.desktop:Exec=/opt/kde3/bin/startkde
kde.desktop:TryExec=/opt/kde3/bin/startkde
kde.desktop:Name=KDE3
twm.desktop:Exec=twm
twm.desktop:TryExec=twm
twm.desktop:Name=TWM
```

Cuando selecciona una sesión particular en gdm o kdm, éste lista las sesiones en**/usr/share/xsessions** y propone la lista. Cuando el usuario se conecta, se escribe su elección como sesión por defecto en su archivo **.dmrc**; luego se ejecuta el comando correspondiente a la línea **Exec**.

2. startx

Si la sesión actual se encuentra en modo comando y tiene instalado el servidor X, puede usar el comando **startx** para iniciar una sesión del servidor X y el entorno asociado.

```
$ startx
```

Cuando el sistema no encuentra ningún archivo de configuración, visualizará X, una pantalla gris con un cursor de ratón que no ofrece ninguna acción.

Cuando inicia X con **startx**, el sistema intenta ejecutar en este orden:

- el archivo **$HOME/.xinitrc**;
- **/etc/X11/xinit/xinitrc**.

Dado que es recomendable no tocar el archivo **/etc/X11/xinit/xinitrc** para que se mantenga fiel al estándar definido por su distribución, puede modificar su archivo personal **.xinitrc**. Como son iguales, puede usar el primero como modelo del segundo. Aquí tiene un ejemplo de **.xinitrc** en Mandriva:

```
$ cat xinitrc
#!/bin/sh

if [ "`whoami`" != root ]; then
  xsetroot -solid "#21449C"
fi

exec /etc/X11/Xsession $*
```

La última línea ejecuta Xsession, que va a buscar un gestor de ventanas para iniciar. Pero un simple archivo que contenga una única línea puede bastar para iniciar KDE, siempre en Mandriva:

```
$ cat $HOME/.xinitrc
/usr/bin/startkde
```

En openSUSE:

```
/opt/kde3/bin/startkde
```

Y para lanzar GNOME:

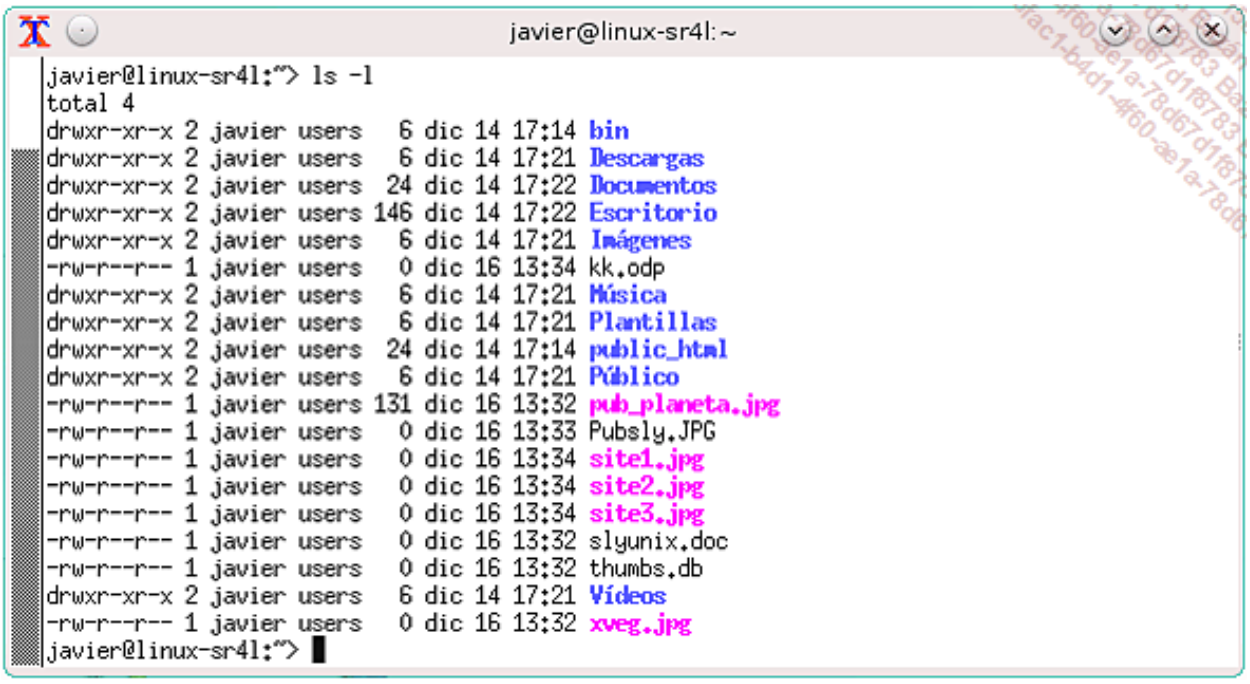
```
exec gnome-session
```

3. Los terminales

Para acceder a línea de comandos desde X, debe utilizar un emulador de terminal. Son numerosos: ¿cuál elegir?

En primer lugar, tenemos al emulador básico del servidor X llamado **xterm**. Sea cual sea su entorno de trabajo X, siempre estará presente. Por eso, el sistema lo inicia por defecto cuando encuentra problemas con el servidor X.

```
$ xterm &
```



Xterm, el terminal virtual clásico

Para configurar Xterm, tiene que utilizar los botones del ratón y la tecla [Ctrl]:

- [Ctrl] + botón izquierdo: opciones principales;
- [Ctrl] + botón derecho: elección de las fuentes de caracteres;
- [Ctrl] + botón del medio: opciones del terminal.

Existen otros emuladores de terminales que podrían gustarle:

- **rxvt**: un terminal muy ligero, pequeñito, que se parece a Xterm pero sin sus menús de configuración. Tenga en cuenta que se verá obligado a utilizar una versión especial llamada urxvt si su distribución es UNICODE.
- **aterm**: un terminal cercano a rxvt, previsto para el gestor Afterstep, que soporta la seudotransparencia del fondo.
- **eterm**: un terminal previsto para sustituir a xterm (como todos los terminales, de hecho) y que fue uno de los primeros en proponer menús. Soporta pestañas en sus últimas versiones.
- **konsole**: el terminal entregado con KDE: soporta perfiles, varios fondos, estilos, transparencia y pestañas. No es el más ligero en tamaño, pero sí uno de los menos voraces en recursos y de los más rápidos.
- **gnome-terminal**: el terminal entregado con GNOME: propone lo mismo que el de KDE, ¿o será al revés?...
- etc.

Puede elegir el terminal que desee, según sus gustos y sea cual sea su entorno de trabajo. Lo importante es lo que hace con él: ejecutar comandos, crear y ejecutar scripts, etc. Sólo recuerde que iniciar un terminal o cualquier otro programa del entorno gráfico tiene como consecuencia consumir recursos, entre otros la memoria. Por ejemplo, iniciar konsole desde GNOME, que, por cierto, funciona muy bien, supone cargar en memoria la ristra de componentes de los que konsole depende.

4. Los gestores de ventanas

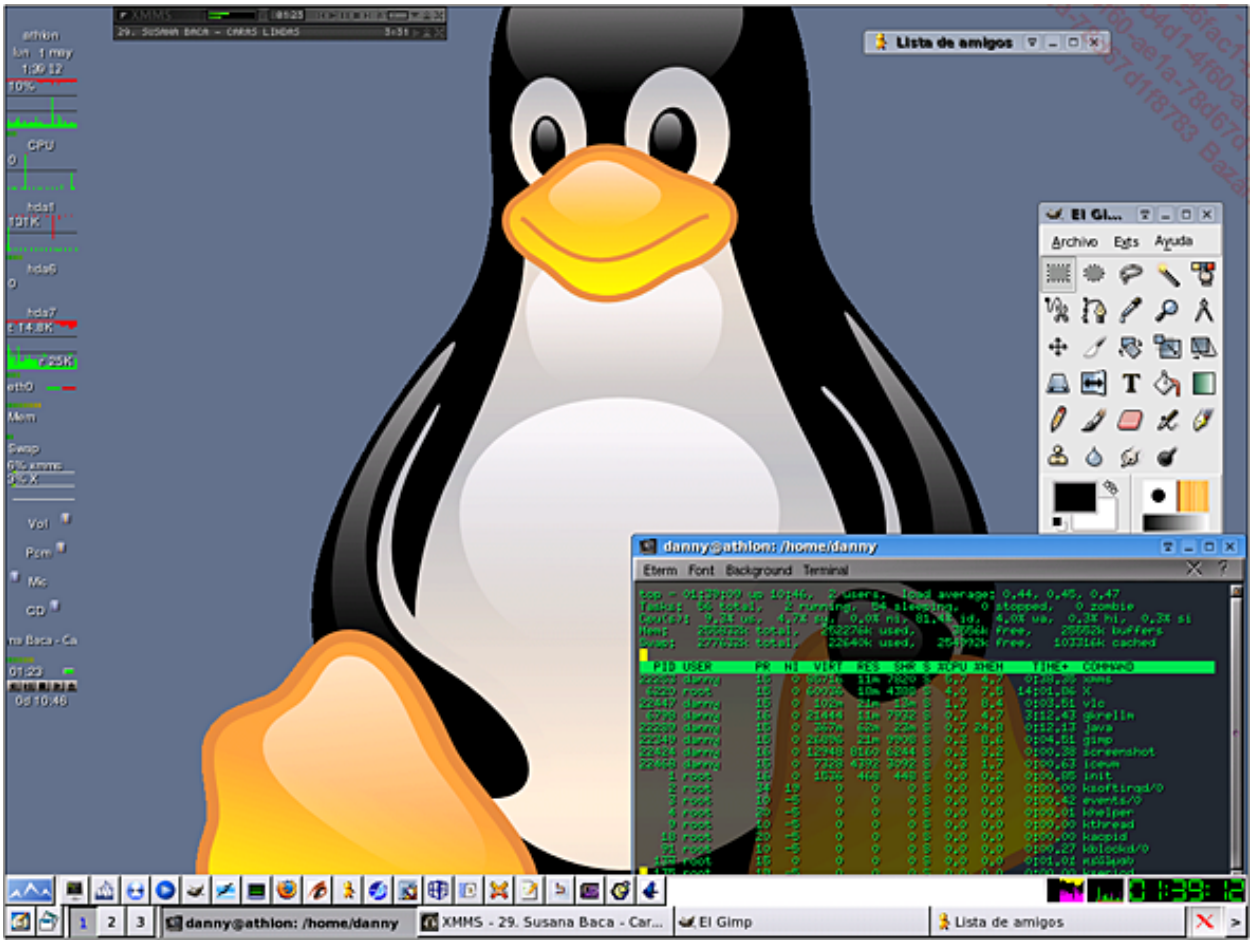
Se ha tratado el principio de los gestores de ventana al comienzo del capítulo. Puede elegir un entorno según sus gustos y necesidades. A continuación le damos una lista de los gestores de ventanas y entornos de escritorio más famosos.

a. twm

El único gestor de ventana oficial de X Window desde la versión X11R4 es TWM: Tom's Window Manager. Es el único que se entrega por defecto, sea cual sea la versión de X. En su versión básica, tal y como pudo ver en la captura al principio de este capítulo, twm es mínimo y está completamente superado por los últimos entornos. Sin embargo, es extremadamente configurable, suministra barras de títulos, iconos, etc. Su configuración se basa en archivos de textos.

b. IceWM

Creado para ser agradable a nivel visual y muy ligero, IceWM presenta la ventaja de sus numerosos temas gráficos, que permiten que se parezca a otros entornos como Windows, OS/2 o lo que quiera. En particular se utiliza IceWM por defecto con los eeePC de Asus (los que se ejecutan en Linux). Su configuración utiliza archivos de texto, aunque cuenta con varios paneles de controles.



IceWM. Captura de la licencia GPL proveniente de Wikimedia Commons

c. fvwm

fvwm es un derivado de twm. La última versión oficial, la 1.24, fue publicada en 1994, y desde entonces han ido apareciendo otras versiones más evolucionadas, como la fvwm95, que se caracterizaba por imitar la barra de las tareas y la estética de las ventanas de Windows 95 y Windows 98. Es altamente configurable y está en la génesis de otros gestores como Afterstep (previsto para agrupar a NextStep), Xfce y Enlightenment. Bien configurado (archivos de texto y aplicaciones gráficas), es espléndido y ligero. Todavía se utiliza hoy en día y la mayoría de los editores de distribuciones sigue publicándolo y dando asistencia.

d. CDE

CDE (*Common Desktop Environment*) es un entorno de escritorio basado en Motif que procede de

openVMS, de HP. Dado que este entorno fue propuesto por The Open Group como estándar, hubiera podido convertirse en común a todos los Unix (de hecho, Solaris, HPUX y True64 lo instalan por defecto). Pero al ser software propietario y haberse liberado Motif demasiado tarde, las plataformas libres se volcaron en los demás, en particular KDE y GNOME.

e. WindowMaker

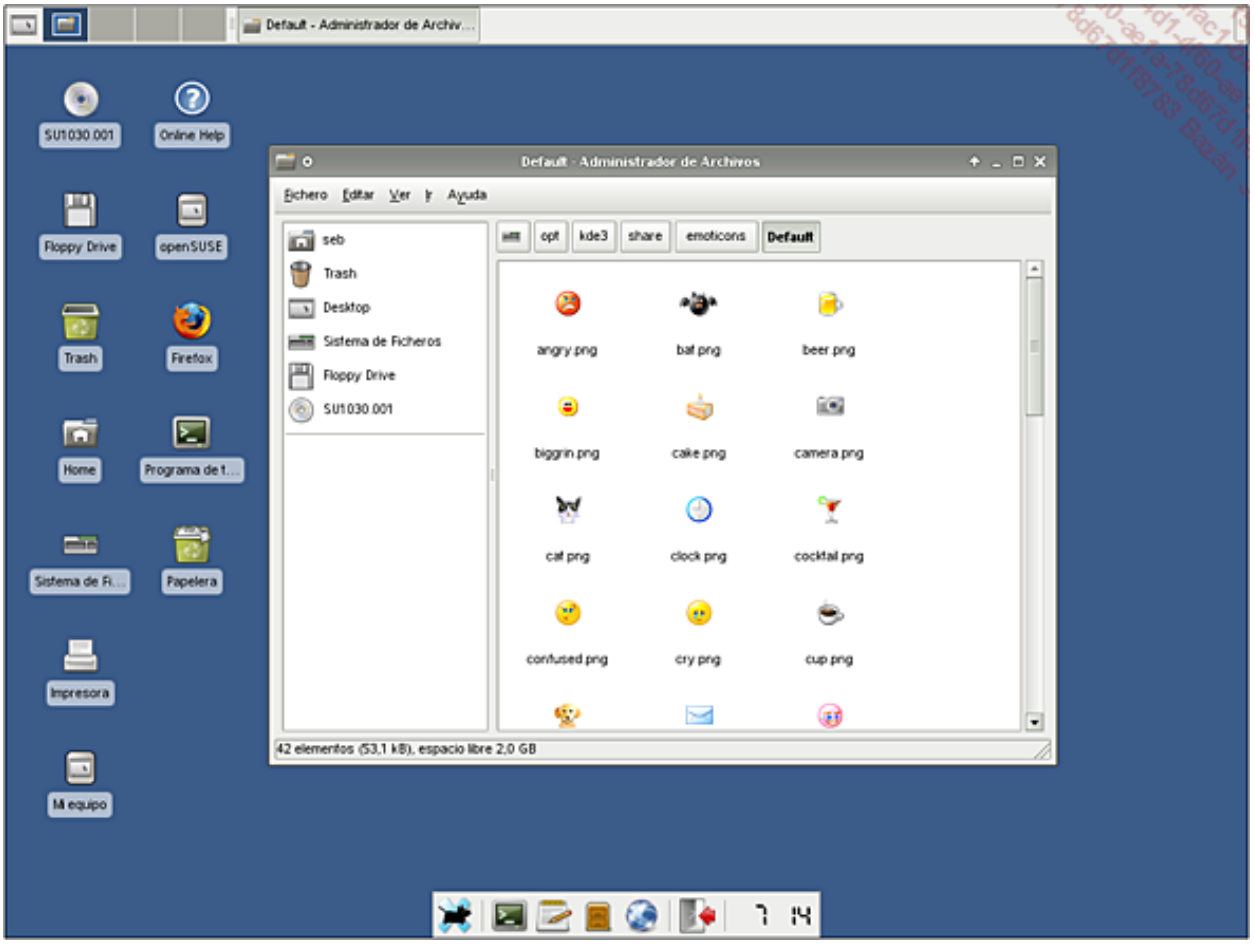
WindowMaker, del que mostramos una captura al principio de este capítulo, es un gestor de ventanas muy evolucionado, desarrollado para parecerse a Nextstep y funcionar con GNUstep, el entorno compatible OpenSTEP de las máquinas NeXT de Steve Jobs. Originalmente se trataba de una versión mejorada de AfterStep. Es estable, fiable, muy bien acabado, fácil de configurar y relativamente ligero.

f. Enlightenment

Enlightenment es el gestor de ventanas de referencia para muchos geeks y nerds. Es totalmente configurable, soporta la ejecución de scripts y es modular. Funciona tanto en pequeñas configuraciones como en grandes servidores. La versión actual, la 16 (actualmente 16.999) existe desde hace varios años, y la versión 17 es la eterna promesa, o el "vaporware" del software libre, con la diferencia de que saldrá realmente un día: cuando esté lista. Será entonces un verdadero entorno de escritorio (Desktop Shell).

g. Xfce

Frente a los mastodónticos KDE y GNOME, las máquinas que disponen de pocos recursos tienen la alternativa Xfce. Fue desarrollado desde esta óptica: un entorno de escritorio completo y muy ligero, basado en las librerías GTK, pero sin consumir recursos de manera excesiva. Funciona, por lo tanto, muy rápidamente con 64 MB de memoria y se entrega con varias aplicaciones, entre las cuales se cuentan un gestor de archivos (Thunar), un editor de texto, etc. En resumen: lo mínimo para trabajar. El usuario cuenta con muchas aplicaciones disponibles para este entorno.



h. KDE y GNOME

KDE y GNOME: se presentan al principio de capítulo. La elección es excelente si su máquina dispone de al menos 256 MB (512 para funcionar cómodamente) y si desea tener un entorno comparable con el de Windows y Mac OS.

i. Los demás

Existen decenas de gestores de ventanas, entre los cuales se cuentan, por ejemplo, Compiz Fusion, que representa probablemente el futuro con sus efectos 3D espléndidos y compuestos. Entre los gestores de ventanas más notables encontrará:

- Afterstep
- Compiz Fusion
- Fluxbox
- Openbox
- Metacity
- Blackbox
- Ion
- Wmii
- etc.

5. Exportar las ventanas

Como pudo ver durante su presentación, X funciona en modo cliente-servidor. Por defecto, el cliente se conecta al servidor X local y se visualiza en la pantalla principal del usuario. Sin embargo, teóricamente se puede exportar la visualización hacia cualquier servidor X.

Imaginemos dos servidores X:

- uno en 192.168.1.60;
- otro en 192.168.1.70.

Quiere iniciar desde el primero el programa **xcalc** hacia el segundo.

Para ello, debe autorizar al segundo servidor X a recibir peticiones exteriores:

- la opción **-nolisten tcp** debe estar deshabitada mediante X o XDM;
- tiene que autorizar la conexión con el comando **xhost**.

El comando **xhost** permite controlar los permisos de acceso al servidor X. Su sintaxis básica toma un más "+" o un menos "-" para autorizar o no de manera global (todo el mundo) las conexiones al servidor X.

```
$ xhost +
access control disabled, clients can connect from any host
$ xhost -
access control enabled, only authorized clients can connect
```

Después de los signos, puede precisar un nombre de anfitrión o de login. En el segundo servidor X, debe autorizar el primero:

```
$ xhost +192.168.1.60
192.168.1.60 being added to access control list
```

Desde la primera máquina tiene dos maneras de exportar la visualización hacia la segunda:

- solamente el programa con el parámetro `–display`;
- todos los programas exportando una nueva variable **DISPLAY**.

En los dos casos tiene que precisar, además de la dirección, el número de servidor y si es preciso de pantalla. Se asigna un identificador a cada servidor X. Si es el único, es el cero (0); si es el segundo, 1, y así sucesivamente. Las ”coordenadas” del servidor X, si está solo, son **192.168.1.70:0**. Si se han acoplado varias pantallas, se identifican de manera numérica: 0.0,1,2, etc. Si no se precisa ninguna pantalla, se elige la primera por defecto. Por lo tanto una coordenada completa es**192.168.1.70:0.0**.

En el primer caso, proceda de este modo:

```
$ xcalc --display 192.168.1.70:0
```

En el segundo caso, exporte la variable **DISPLAY** con el nuevo valor:

```
$ export DISPLAY=192.168.1.70:0$
xcalc
```

La política de seguridad por defecto consiste en impedir el acceso al servidor X incluso si se precisa xhost+. En este caso, y esperando que lo permita el servidor ssh, pase por ssh con el parámetro -X.

6. Los recursos de una aplicación X

a. Modificar la apariencia de un programa

Cada programa X invoca recursos del servidor X Window para poder funcionar. En concreto, necesita del servidor las fuentes de caracteres, la forma de los botones, los colores, etc. El archivo Xresources de la configuración de Xdm es un buen ejemplo.

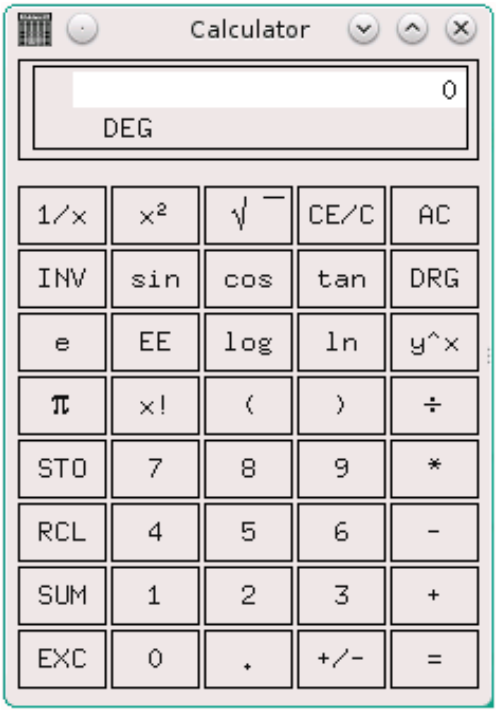
Es posible establecer los recursos que puede utilizar un programa X, excepto con las aplicaciones KDE o GNOME, que no están sujetas a dicha limitación.

Cada programa X admite varios parámetros estándares en la línea de comandos, entre los cuales se encuentran:

- -**bg**: color de fondo, recurso *background.
- -**bd**: color de borde, recurso *borderColor.
- -**fg**: color de primer plano, recurso *foreground.
- -**fn**: fuente principal, recurso *font.
- -**geometry**: tamaño y posición de la ventana, recurso *TopLevelShell.geometry.
- -**title**: título de la ventana, recurso *title.
- etc.

Xcalc puede iniciarse en dos modos: TI-30 y HP (notación polaca invertida).

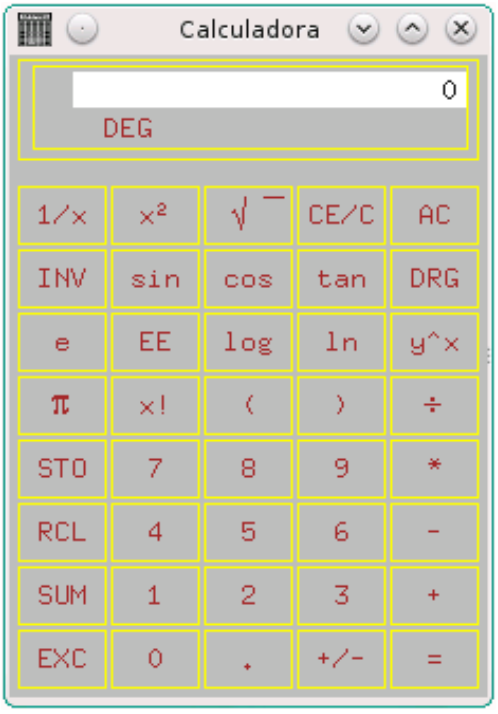
En el ejemplo siguiente se propone modificar la visualización de la aplicación **xcalc**. Primero, una captura de la calculadora antes de las modificaciones.



La calculadora X por defecto

Y ahora tenemos el mismo resultado con un cambio de los parámetros por defecto: título "Calculadora", fondo gris, color principal marrón, bordes amarillos:

```
$ xcalc -title "Calculadora" -bg grey -fg brown -bd yellow &
```



La misma calculadora, modificada mediante la línea de comandos

b. Editar los recursos

El comando **xprop** permite conocer todas las propiedades de una ventana X.

- Primero inicie `xcalc`.

```
$ xcalc &
```

Luego inicie `xprop` desde un terminal.

```
$ xprop
```

El cursor del ratón se transforma en una cruz. Haga clic en la ventana de xcalc. Tendría que obtener una salida de este tipo:

```
_NET_WM_ICON_GEOMETRY (CARDINAL) = 649, 984, 133, 20
WM_PROTOCOLS(ATOM): protocols WM_DELETE_WINDOW
WM_STATE(WM_STATE):
              window state: Normal
              icon window: 0x0
_NET_WM_STATE(ATOM) =
_NET_WM_ALLOWED_ACTIONS(ATOM) = _NET_WM_ACTION_MOVE,
_NET_WM_ACTION_RESIZE, _NET_WM_ACTION_MINIMIZE,
_NET_WM_ACTION_SHADE, _NET_WM_ACTION_MAXIMIZE_VERT,
_NET_WM_ACTION_MAXIMIZE_HORZ, _NET_WM_ACTION_FULLSCREEN,
_NET_WM_ACTION_CHANGE_DESKTOP, _NET_WM_ACTION_CLOSE
_KDE_NET_WM_FRAME_STRUT(CARDINAL) = 5, 5, 19, 5
_NET_FRAME_EXTENTS(CARDINAL) = 5, 5, 19, 5
_NET_WM_DESKTOP(CARDINAL) = 0
WM_CLIENT_LEADER(WINDOW): window id # 0x3e00026
WM_LOCALE_NAME(STRING) = "es_ES.UTF-8"
_KDE_NET_WM_USER_CREATION_TIME(CARDINAL) = 1321633138
WM_CLASS(STRING) = "xcalc", "XCalc
"WM_HINTS(WM_HINTS):
              Client accepts input or input focus: True
              Initial state is Normal State.
              bitmap id # to use for icon: 0x3e00001
WM_NORMAL_HINTS(WM_SIZE_HINTS):
              program specified size: 226 by 308
              window gravity: NorthWest
WM_CLIENT_MACHINE(STRING) = "slyserver"
WM_COMMAND(STRING) = { "xcalc" }
WM_ICON_NAME(STRING) = "Calc"
WM_NAME(STRING) = "Calculator"
```

En la información devuelta, la línea en negrita indica el nombre (xcalc o XCalc) que se deberá utilizar para acceder a los recursos de la aplicación X. Suele ser el nombre real del programa.

Un recurso se presenta con la forma:

```
class*resource: value
instance*resource: value
class.resource: value
instance.resource: value
```

- **class** representa todas las invocaciones de un mismo programa. La clase XCalc representa todas las instancias de XCalc, por lo tanto si ejecuta n veces el programa xcalc los recursos de tipo xcalc.xxxx serán comunes a todos los xcalc iniciados.
- **instancia** es el nombre de una instancia de un programa particular.

El manual de xcalc facilita en las secciones CUSTOMIZATION y WIDGET HIERARCHY los recursos y valores particulares aceptados. Xcalc dispone de los modos hp y ti. En modo ti, por ejemplo, los recursos son del tipo:

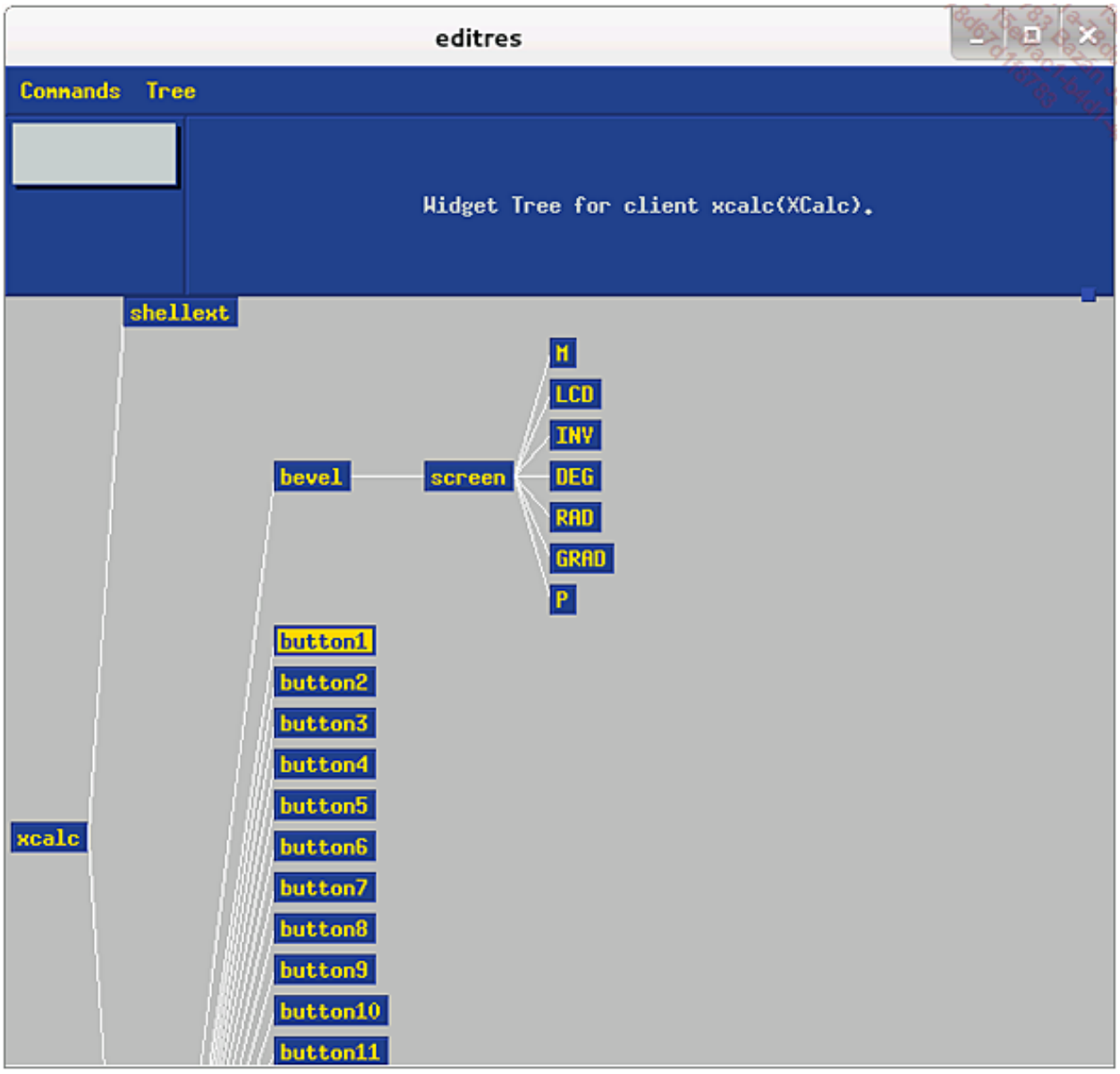
```
XCalc.ti.resource: value
```

Pero si sustituye ti o hp por un asterisco "*", la definición del recurso se aplicará al conjunto. Los propios recursos se pueden descomponer en subrecursos.

El comando **appres** toma como parámetro una instancia o una clase y devuelve todos los recursos asociados. Es ideal para guardar los recursos de una clase ante de modificarlos. Sin embargo, visualiza también los recursos aplicables a XCalc, que por defecto empiezan por "*":

```
$ appres XCalc | grep -i xcalc
...
XCalc*ti.button13.fromVerde:            button8
XCalc*ti.button30.translations:         #override
<Btn1Down>,<Btn1Up>:subtract()unset()
XCalc*ti.button30.background:           rgb:e/d/c
XCalc*ti.button30.borderColor:          rgb:9/8/7
XCalc*ti.button30.foreground:           gray5
XCalc*ti.button30.fromVerde:            button25
XCalc*ti.button30.fromHoriz:            button29
XCalc*ti.button30.label:                -
XCalc*ti.button30.displayList:          foreground rgb:a/9/8;segments
8,-4,-9,-4,-4,-9,-4,8;draw-arc -14,-14,-4,-4,270,90
XCalc*ti.button14.label:                ln
XCalc*ti.button14.translations:         #override
<Btn1Down>,<Btn1Up>:naturalLog()unset()
XCalc*ti.button14.fromHoriz:            button13
XCalc*ti.button14.fromVerde:            button9
XCalc*ti.Command.background:            rgb:c/d/e
XCalc*ti.Command.borderColor:           rgb:8/9/a
XCalc*ti.Command.shapeStyle:            roundedRectangle
XCalc*ti.Command.foreground:            gray5
XCalc*ti.Command.displayList:           foreground rgb:a/b/c;segments
8,-4,-9,-4,-4,-9,-4,8;draw-arc -14,-14,-4,-4,270,90
XCalc*ti.backgroundPixmap:
gray3?foreground=gray70&background=gray85
XCalc*bevel.screen.GRAD.fromHoriz:      RAD
XCalc*bevel.screen.GRAD.fromVerde:      LCD
XCalc*bevel.screen.INV.fromVerde:       LCD
XCalc*bevel.screen.INV.verdeDistance:   4
XCalc*bevel.screen*INV.verdeDistance:   2
XCalc*bevel.screen.P.label:             ()
XCalc*bevel.screen.P.fromHoriz:         GRAD
XCalc*bevel.screen.P.fromVerde:         LCD
XCalc*bevel.screen.P.horizDistance:     2
XCalc*bevel.screen.LCD.label:           88888888888
XCalc*bevel.screen.LCD.fromHoriz:       M
XCalc*bevel.screen.LCD.horizDistance:   4
XCalc*bevel.screen.LCD.verdeDistance:   2
XCalc*bevel.screen*LCD.foreground:      gray20
XCalc*bevel.screen.DEG.fromHoriz:       INV
XCalc*bevel.screen.DEG.fromVerde:       LCD
XCalc*bevel.screen.DEG.horizDistance:   1
XCalc*bevel.screen.Label.horizDistance: 4
XCalc*bevel.screen.Label.verdeDistance: 2
XCalc*bevel.screen.Label.internalHeight:1
XCalc*bevel.screen.Label.internalWidth: 1...
```

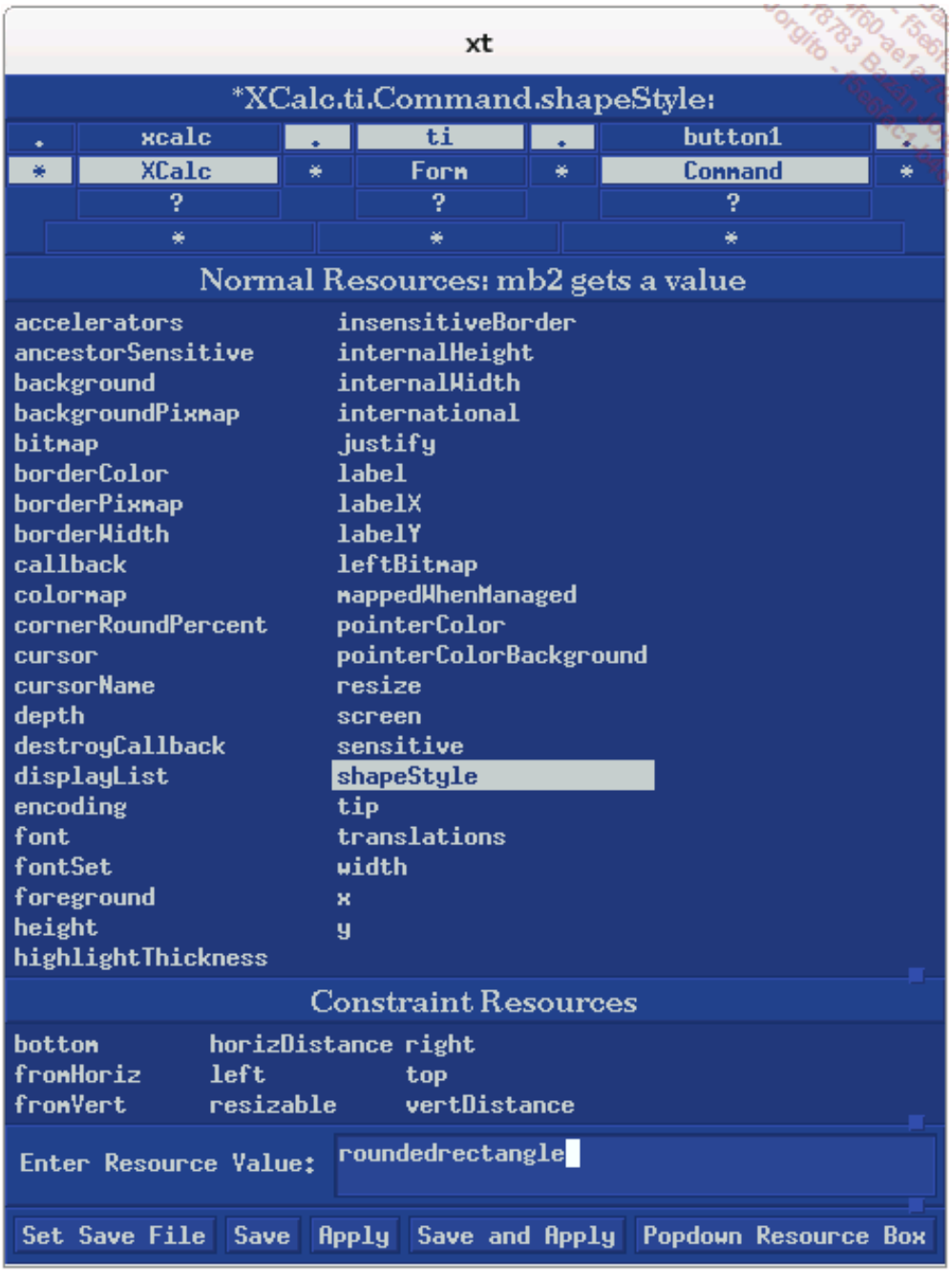
Existe un editor simple para editar los recursos de un programa y guardar los parámetros. Se llama**editres**. Inícielo desde el menú **Commands**, seleccione **Get Tree**. Luego haga clic en la ventana X, la de Xcalc.



editres: editor de recursos X

Para modificar un valor sobre la marcha, haga clic en un recurso, por ejemplo **button1**, luego en**Commands** sobre **Show Resource Box**. En este cuadro elija **background**, teclee **red** y luego haga clic en **Apply**. El fondo del botón **1/x** se vuelve rojo.

En el menú **Tree**, seleccione **Show class name**. Haga clic en uno de los **Command** y luego vuelva al cuadro de los recursos. Va a modificar todos los recursos para todas las clases **Command**: haga clic arriba en **XCalc**, *****, **ti**, **.**, **Command**, luego en el cuadro central en **shapeStyle** (arriba debe aparecer **.XCalc*ti.Command.shapeStyle**). Teclee **roundedRectangle**, luego **Apply**. ¡Todos los botones se han redondeado!



Edición de un recurso de xcalc con editres

Guardar el cambio con los botones **Set Save File** y luego **Save**.

```
$ cat Xcalc.res
.XCalc*ti.Command.shapeStyle: roundedRectangle
```

c. xrdb

Puede guardar sus ajustes en un archivo cualquiera, pero X intenta leer el archivo $HOME/.Xdefaults en el momento de abrir la sesión. Se utiliza el comando **xrdb** para editarlo. Admite los parámetros siguientes:

- **ninguno**: el nuevo archivo sobreescribe todos los antiguos ajustes;
- **-merge**: se añade el archivo a los ajustes anteriores;
- **-remove**: suprime un recurso. Sin nada: suprime todos los recursos.

Supongamos el contenido siguiente:

```
$ cat .Xdefaults
XCalc*title: Calculadora
XCalc*ti.Command.shapeStyle: roundedRectangle
XCalc*ti.Command.background: cyan
```

- El título de la ventana será Calculadora.
- Los botones tendrán las esquinas redondeadas.
- El fondo de los comandos (los botones) serán azul celeste.

Cargue los ajustes de este modo:

```
$ xrdb -merge .Xdefaults
```

Y mire el resultado:

```
$ xcalc
```

Xcalc modificado empleando los archivos de recursos

Accesibilidad

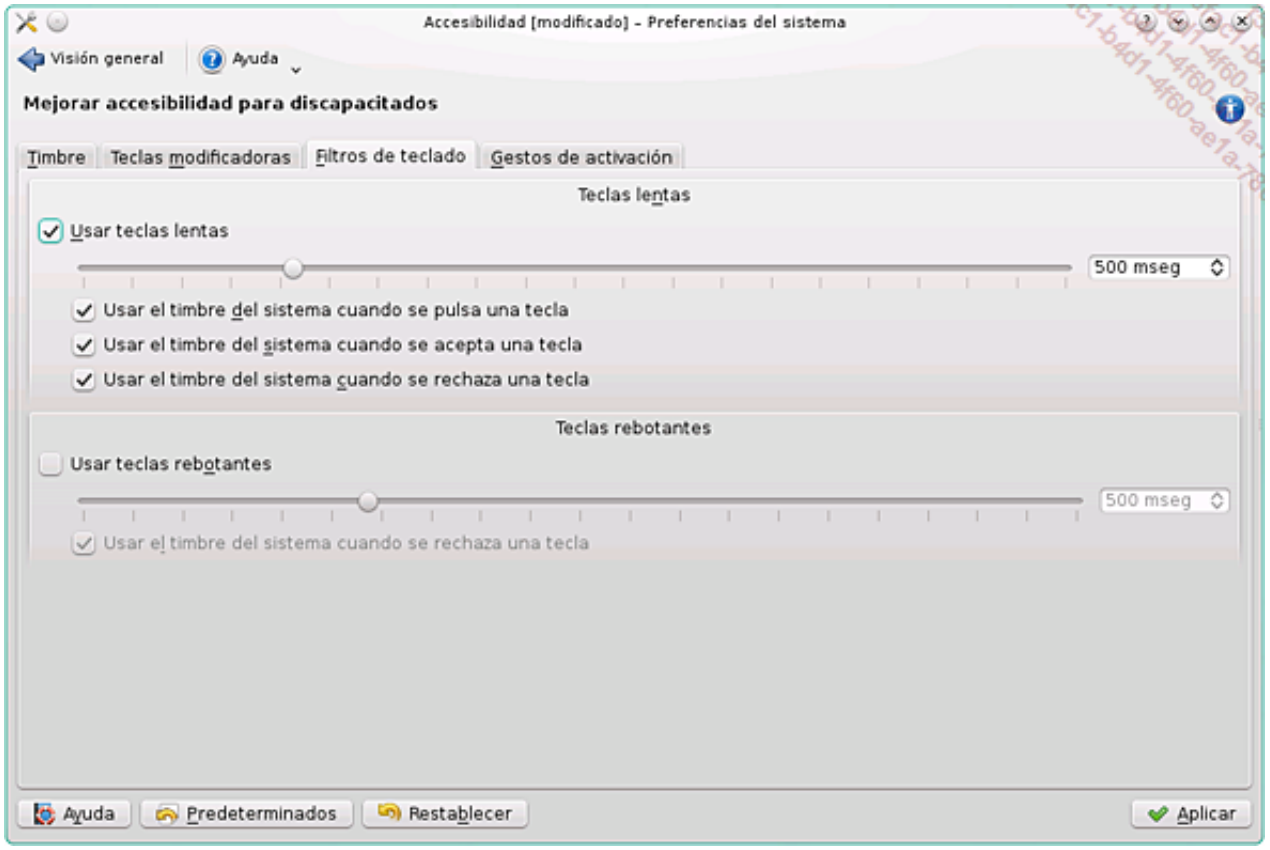
Si bien la mayoría de usuarios no tienen problemas para leer una pantalla, escribir con un teclado y utilizar un ratón, otros pueden padecer trastornos que les impidan trabajar de manera normal. Según la minusvalía, Linux y sus herramientas proponen varios medios para ayudar a estas personas.

Esta sección repasa algunos de los medios disponibles para ayudar a los minusválidos, en particular con discapacidades motrices o visuales.

1. Asistencia al teclado y el ratón

Un simple ajuste del teclado puede ayudar une persona con problemas motores. Por ejemplo, una persona lenta va a dedicar mucho tiempo a buscar una tecla, con el riesgo de equivocarse; luego apretará demasiado tiempo la tecla deseada, lo que tendrá como efecto repetir el carácter tecleado varias veces.

El módulo AccessX del entorno gráfico permite muchos ajustes. Son accesibles desde los centros de configuración de los entornos gráficos. En KDE4, por ejemplo, el módulo **Accesibilidad** de la pestaña**General** de la configuración del sistema propone modificar numerosos ajustes por defecto, entre los cuales se encuentra el funcionamiento del teclado. Por ejemplo, en la pestaña **Filtros de teclado**, marque **Usar teclas lentas**. Si una persona quiere efectuar una acción, no se tendrá en cuenta hasta que se cumpla el plazo elegido (0,5 segundos por defecto). Debajo, **Usar teclas rebotantes**espera el plazo indicado antes de repetir una tecla en caso de apretar durante demasiado tiempo.

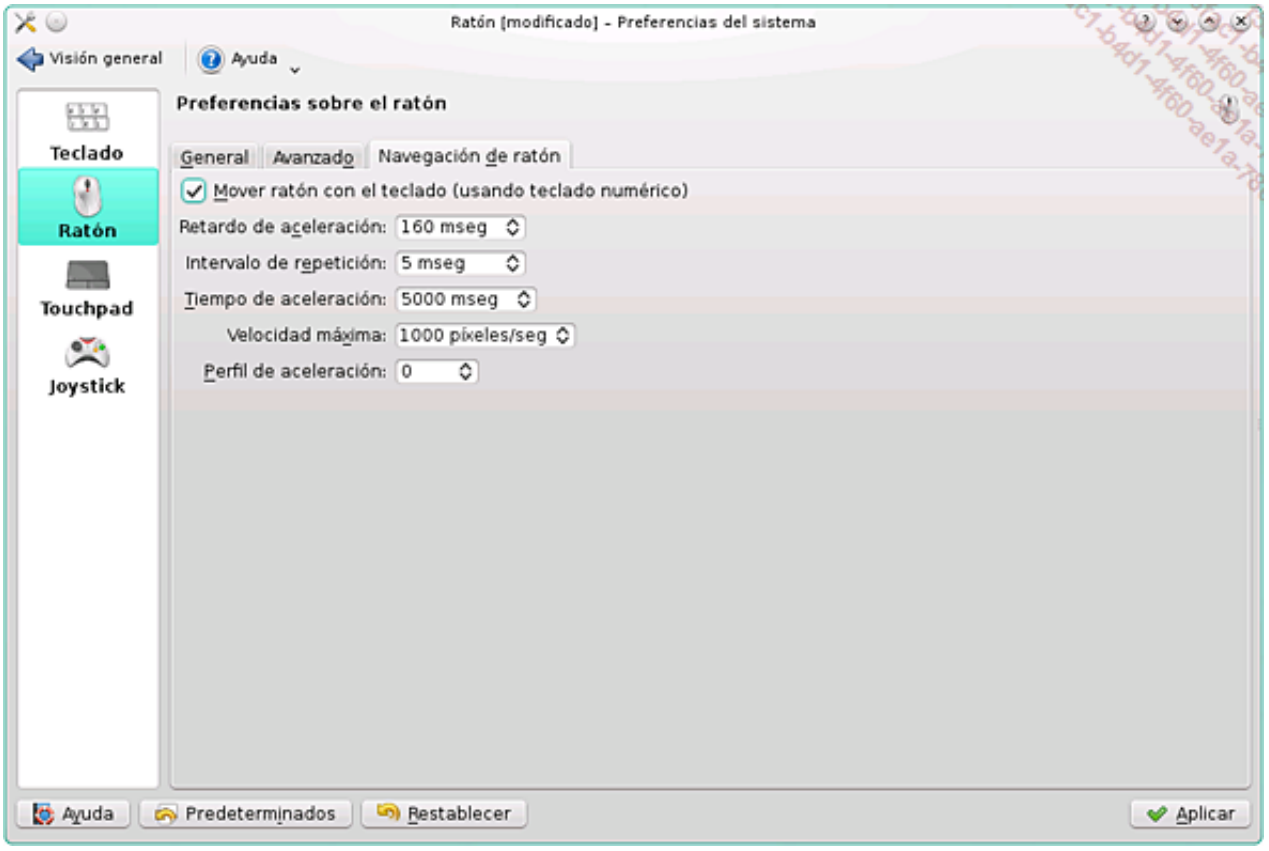


Los parámetros de accesibilidad para el teclado de KDE4

En el mismo sentido, puede activar un soporte visual o auditivo según las acciones efectuadas. Al presionar una tecla puede provocar un bip, el parpadeo de la pantalla. Los movimientos del ratón pueden contribuir a la mejora de la accesibilidad.

Como no todo el mundo puede utilizar el ratón, es posible desplazar el cursor en la pantalla usando el teclado. Lea la documentación de su entorno para la activación de este modo operativo. Siempre

en KDE4, en el módulo **Teclado y Ratón**, pestaña **Navegación de ratón**, si marca **Mover ratón con el teclado**, podrá mover el cursor del ratón con las teclas de dirección del teclado numérico. Las teclas de 1 a 9, salvo la 5, desplazan el cursor en la dirección deseada. La tecla 5 simula un simple clic (5 dos veces rápidamente simula un doble clic). Es posible cambiar de botón, mantenerlo, etc.



Utilización de las teclas del teclado numérico para desplazar el ratón

En cuanto al ratón, si dispone de un modelo adecuado, puede activar el perfil para zurdos, que va a invertir los botones e incluso el sentido de la rueda.

Finalmente, el teclado virtual puede ayudar mucho en caso de dificultad de uso del teclado físico: a partir de ese momento, puede utilizar un teclado de software con el ratón o una pantalla táctil. Basta con hacer clic en las diferentes teclas para utilizarlo como un teclado ordinario. El de KDE se llama **kvkbd** pero X Window propone uno por defecto llamado **xvkbd** y Gnome propone GOK.

El teclado virtual kvkbd permite evitar el teclado físico.

2. Asistencia visual y auditiva

Las personas con problemas de visión apreciarán la posibilidad de utilizar notificaciones visuales, como el parpadeo de la pantalla. Además, todos los entornos de escritorio dignos de este nombre, como KD3, KDE4 o GNOME (por ejemplo), proponen modificar la visualización:

- **Colores**: aumentando por ejemplo los contrastes o con temas según la representación de los colores (daltonismo).
- **Estilos**: modificando el tamaño de varios elementos visuales, como los botones, las casillas, los campos, etc.
- **Fuentes de caracteres**: elegiendo una fuente adaptada, de mayor tamaño, de un estilo determinado.
- **Temas**: existen temas específicos para los invidentes que modifican el conjunto de los ajustes anteriores de una vez.

Hay herramientas sencillas que permiten ampliar zonas de la pantalla con una lupa virtual. KDE propone Kmagnifier. También es posible asociar acciones (teclas del teclado o del ratón) a la lupa para tener un acceso más rápido.

Las personas invidentes también pueden trabajar con Linux, ya que existen soluciones para leer una pantalla en braille. Existen varias soluciones que soportan el braille, entre otras brltty, para el traspaso de la consola a un lector de braille, y orca, producto estrella en Gnome, que proporciona el soporte del braille en el entorno Gnome, pero también voces sintéticas y la accesibilidad a la pantalla para las personas ciegas.

Para terminar, existen varios medios para que el ordenador "hable". Productos tales como espeak (sintetizador muy "computerizado") y sobre todo mbrola con freetts dan resultados muy cercanos a una voz humana.

Particionamiento avanzado RAID

1. Definiciones

El **RAID** (*Redundant Array of Inexpansive Disks*) ha sido definido por la Universidad de Berkeley en 1987 con el doble objetivo de reducir los costes y aumentar la fiabilidad del almacenamiento de datos. El objetivo es compaginar varios pequeños discos físicos independientes en una matriz (array: tabla, conjunto, fila, matriz) de discos cuya capacidad supera la del SLED (*Single Large Expensive Drive*). Una matriz aparece como una unidad lógica de almacenamiento único.

El **MTBF** (*Mean Time Between Failure* - intervalo entre fallos) del conjunto es igual al MTBF de un disco individual dividido por el número de discos en el conjunto y por lo tanto, teóricamente, una solución RAID puede no ser la adecuada en sistemas críticos. Afortunadamente, un sistema RAID es tolerante a los fallos gracias a que almacena de manera redundante su información según varios métodos:

- **RAID-0**, llamado **stripe mode**: al menos dos discos forman un único volumen. En principio, los dos discos tienen el mismo tamaño. Se fraccionará y efectuará cada operación de lectura/escritura en cada uno de los discos. Por ejemplo, se escribirán 4 KB en el disco 0, 4 KB en el disco 1, 4 KB en el disco 2, luego 4 KB en el disco 0, etc. Así, se aumentan las prestaciones, ya que se efectúan en paralelo las operaciones de lectura y escritura sobre los discos. Si N es el número de discos y P la velocidad de transferencia, la velocidad de transferencia del volumen RAID es, en principio, próxima a N*P mbps. El RAID-0 no tiene ninguna redundancia. En caso de fallo de uno de los discos, es probable que se pierda el conjunto de datos.
- **RAID-1**, llamado **mirroring**: primer modo redundante. Se puede utilizar a partir de dos discos o más con posibles discos de emergencia (*Spare Disk*). Se duplica cada información escrita en un disco sobre los demás. Si N-1 discos del RAID caen, los datos siguen intactos. Si hay un disco de emergencia, en caso de fallo, se reconstruye automáticamente y se sustituye el disco que falla. Las prestaciones en escritura pueden ser malas: escritura en N discos al mismo tiempo, con el riesgo de saturar el controlador de disco y el bus. Las prestaciones en lectura son buenas, porque RAID emplea un algoritmo que puede leer los datos en cada disco (ya que son idénticos).
- **RAID-5**: RAID con algoritmo distribuido de paridad. Es el modo más utilizado, ya que ofrece la mejor relación entre el número de discos, el espacio disponible y la redundancia. Hacen falta al menos tres discos con posibles discos de emergencia. Hay paridad en cada uno de los discos. El tamaño final es el de N-1 discos. El RAID-5 sobrevive a un fallo de disco. En este caso, si hay un disco de emergencia, será reconstruido automáticamente. Las prestaciones en lectura son equivalentes a las del RAID-0, mientras que en escritura dependen del algoritmo empleado, así como de la memoria de la máquina.

2. Precauciones y consideraciones de uso

a. Disco de emergencia

Un disco de emergencia (*Spare Disk*) no forma parte integrante de una matriz RAID mientras no se averíe un disco. Si eso sucede, se marca el disco como defectuoso y el primer disco Spare toma el relevo. Pase lo que pase, conviene cambiar el disco averiado lo antes posible y volver a construir el RAID.

b. Disco averiado

Un disco averiado (*Faulty Disk*) es un disco que ha sido reconocido como defectuoso o como erróneo por el RAID. En este caso, RAID utiliza el primer disco SPARE para reconstruir su matriz. Los discos faulty pertenecen siempre a la matriz, pero están desactivados.

c. Boot

No se debe ubicar en una matriz RAID la partición boot (la que contiene el núcleo, la configuración del bootloader, los archivos imágenes de discos): el cargador del sistema operativo es incapaz de montar particiones RAID (la próxima versión del GRUB sí que podrá).

d. Swap

Se puede instalar una partición de intercambio (swap) sobre un RAID, pero no suele ser útil en los casos comunes. Linux es capaz de equilibrar el uso del swap sobre varios discos/particiones individuales. En este caso, declare n swaps en **/etc/fstab** con la misma prioridad.

```
/dev/sda2 swap swap defaults,pri=1 0 0
/dev/sdb2 swap swap defaults,pri=1 0 0
/dev/sdc2 swap swap defaults,pri=1 0 0
```

Sin embargo, en caso de necesidad de alta disponibilidad, es posible la swap en el RAID.

e. Periféricos

El sistema reconoce una matriz RAID como un periférico de tipo bloque, al igual que cualquier disco físico. Así, un RAID puede estar constituido por discos, particiones (en general, se crea una única partición en cada disco). El bus no tiene importancia alguna: puede construir una matriz RAID con discos SCSI e IDE mezclados. Asimismo, se puede construir un RAID sobre otras matrices RAID, por ejemplo RAID-0+1 (2x2 discos sobre RAID-1, formando sobre RAID-0 una nueva matriz las dos resultantes). Los periféricos RAID tienen la forma:

```
/dev/md0
/dev/md1
```

f. IDE y SATA

Si los discos IDE han sido mucho tiempo el SCSI del pobre (hardware de menor calidad, lentitud, falta de fiabilidad), hoy por hoy esto ya no es cierto. Los últimos modelos son totalmente idénticos a los discos SCSI, si exceptuamos el controlador. De hecho, puede montar configuraciones RAID en IDE a un coste razonable. Sin embargo, hay que recordar una regla: **un solo disco IDE por bus IDE**.

En la práctica, esto corresponde a poner un disco por bus, nada más. La razón estriba en que un bus IDE sobrevive al fallo en un disco, pero si lo que falla es el bus mismo se pierden todos los discos conectados al bus y con ellos la matriz RAID. La compra de tarjetas IDE adicionales (a bajo precio) permite compensar el problema de fiabilidad (dos discos por tarjeta).

El principio es el mismo que para los discos SATA. En teoría es más simple, ya que sólo puede conectar un disco por cable. Sin embargo, si lee el manual de su placa base o el del controlador, le recordarán que a menudo los puertos SATA van por parejas. Además, si su placa contiene varios controladores, puede ser una buena idea separar los discos: uno por controlador, o como con IDE, añadir un controlador SATA adicional por PCI o PCI-Express.

g. Hot Swap

- **IDE**: **¡Nunca desenchufe un disco IDE en caliente!** Es la mejor manera de destruir el disco, si ya no fuese el caso, y de destruir el controlador IDE (y, en ocasiones, la placa base o la IDE adicional). El disco IDE no está diseñado para ello.
- **SCSI**: los controladores SCSI no están previstos para el Hot Swap, pero en teoría deberían funcionar de todas maneras, si el disco es idéntico física y lógicamente.
- **SATA**: Se equipara el SATA con un SCSI. La especificación SATA en versión 2 soporta teóricamente el Hot Swap. Sin embargo, la mayoría de los controladores actuales o no implementan o implementan mal esta posibilidad; de ahí que haya riesgos de bug o de

quemar su controlador. Compruébelo en la documentación del fabricante de su placa base (chipset).

- **SCA**: son discos SCSI específicos. Consulte el documento "Software RAID Howto".

3. RAID con mdadm

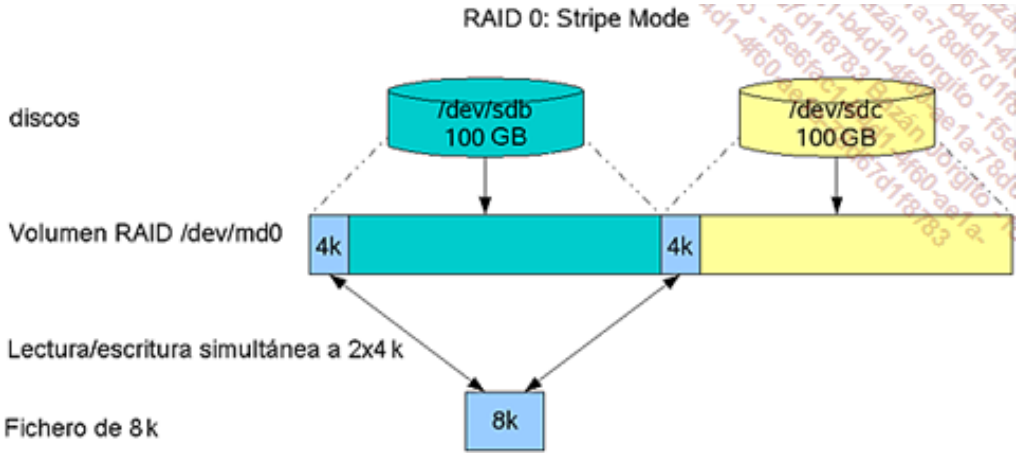
a. Preparación

La herramienta **mdadm** sustituye a las herramientas raidtools de las antiguas distribuciones Linux. Esta herramienta es más sencilla y permite efectuar todas las operaciones. Su archivo de configuración es **`/etc/mdadm.conf`**.

Con el fin de crear matrices RAID, es necesario que las particiones que servirán para crear la matriz sean de tipo **0xFD** (Linux RAID autodetect). Las particiones deben estar físicamente en discos diferentes, pero para hacer pruebas el soporte RAID autoriza particiones en un mismo disco. En ese caso, deberá tener en cuenta que las particiones dispongan del mismo tamaño.

b. Creación

RAID-0



Fundamentos de RAID 0: Striping

O sea, dos particiones **`/dev/sdb1`** y **`/dev/sdc1`**. Va a crear una partición RAID-0, ensamblaje de estas dos particiones.

```
# mdadm --create /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

`--create`	Crea un RAID.
`/dev/md0`	Nombre del archivo de tipo bloque que representa la matriz RAID.
`--level`	Tipo de RAID que se va a crear: `0`, `raid0` y `stripe` para RAID0.

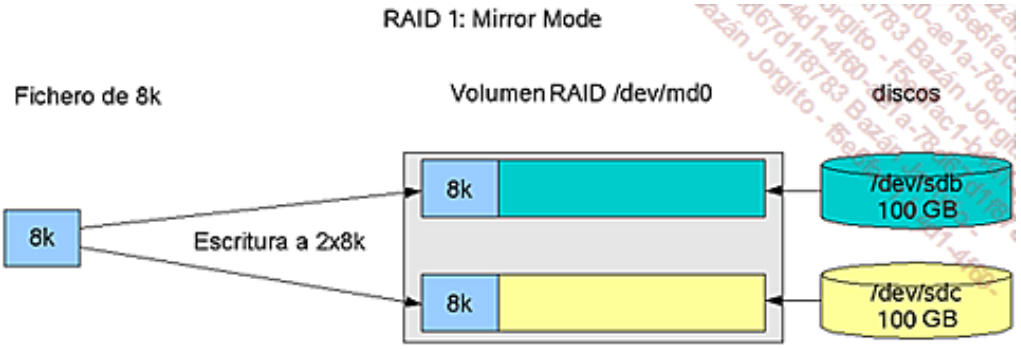
`linear` no es RAID0 (llenado poco a poco).

`--raid-devices`	Número de particiones utilizadas para crear la matriz.
`/dev/sdb1, /dev/sdc1`	Particiones que forman la matriz, según el número indicado en `--raid-devices`.

Sólo queda instalar el sistema de archivos en el disco RAID:

```
# mkfs -t ext4 /dev/md0
```

RAID-1



El principio es el mismo. Esta vez, tendrá que añadir una partición de emergencia **/dev/sdd1**.

```
# mdadm --create /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb1
/dev/sdc1 --spare-devices=1 /dev/sdd1
```

`--level 1`, `mirror` o `raid1` son valores aceptados para un RAID-1.

`--spare-devices` número de discos de emergencia para utilizar.

`/dev/sdd1` partición o particiones de emergencia, según el número indicado en `-spare-devices`.

Luego:

```
# mkfs -t ext4 /dev/md0
```

RAID-0+1

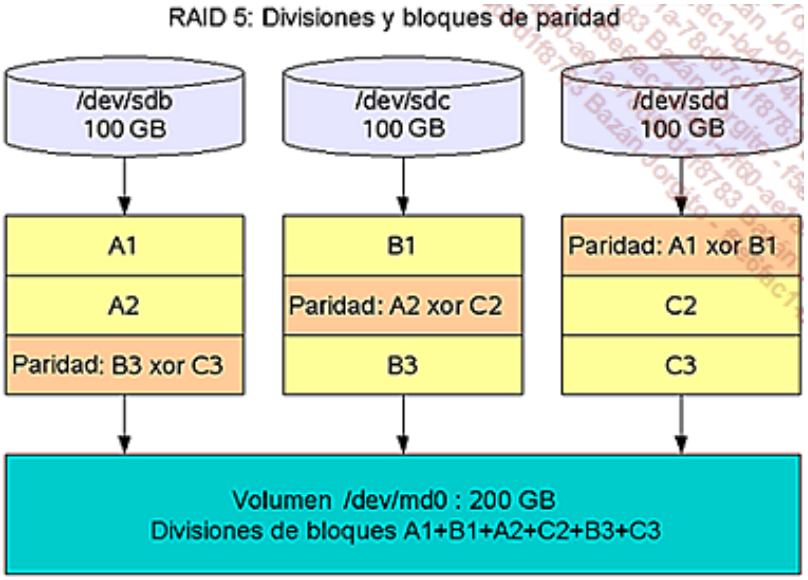
Hacen falta al menos cuatro particiones. Debe crear dos matrices RAID-1, que va a agrupar en una sola matriz RAID-0.

```
# mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb1
/dev/sdc1

# mdadm --create /dev/md1 --level=raid1 --raid-devices=2 /dev/sdd1
/dev/sde1

# mdadm --create /dev/md2 --level=raid0 --raid-devices=2 /dev/md0
/dev/md1
```

Luego:

```
# mkfs -t ext4 /dev/md2
```

RAID5



El RAID va a emplear tres discos de datos /dev/sdb1, /dev/sdc1, /dev/sdd1 y un disco de emergencia /dev/sde1.

```
# mdadm --create /dev/md0 --level=raid5 --raid-devices=3 /dev/sdb1
/dev/sdc1 /dev/sdd1

--spare-devices=1 /dev/sde1
```

Luego se instala el sistema de archivos:

```
# mkfs -t ext4 /dev/md2
```

c. Guardar la configuración

Para facilitar la tarea de la herramienta **mdadm**, puede crear (no es obligatorio) el archivo de configuración **`/etc/mdadm.conf`**. Se puede crear este archivo manualmente, pero la herramienta **mdadm** sabe generarlo. Es preferible hacerlo DESPUÉS de la creación de las matrices RAID.

```
# echo "DEVICE partitions" > /etc/mdadm.conf

# mdadm --detail --scan>> /etc/mdadm.conf
```

4. Estado del RAID

El archivo virtual **`/proc/mdstat`** contiene información sobre el RAID. En este archivo puede ver el detalle de un RAID, en particular si alguno de los volúmenes de la matriz está averiado (Faulty).

```
Personalities: [raid1]
md0: active raid1 hda10[2] hda9[1] hda8[0]
     104320 blocks [2/2] [UU]
```

El comando **watch** permite comprobar un estado en tiempo real:

```
# watch cat /proc/mdstat
```

También puede utilizar **mdadm** con el parámetro `--detail`:

Cabe señalar que, con este último comando, puede obtener muchos más detalles, en particular

```
# mdadm --detail /dev/md0
/dev/md0:
        Versión: 00.90.01
  Creation Time: Mon Jan 23 22:10:20 2006
     Raid Level: raid1
     Array Size: 104320 (101.88 MiB 106.82 MB)
    Device Size: 104320 (101.88 MiB 106.82 MB)
   Raid Devices: 2
  Total Devices: 3
Preferred Minor: 1
    Persistence: Superblock is persistent

    Update Time: Mon Jan 23 22:13:06 2006
          State: clean
 Active Devices: 2
Working Devices: 3
 Failed Devices: 0
  Spare Devices: 1

    Number   Major   Minor   RaidDevice State
       0       3        8        0      active sync   /dev/hda8
       1       3        9        1      active sync   /dev/hda9
       2       3       10       -1      spare         /dev/hda10
           UUID: 90e838b5:936f18c7:39f665d3:d9dad1a9
         Events: 0.4
```

cuáles son los discos ”spare” y "faulty".

5. Simular una avería

Ahora va a simular une avería en **/dev/hda8**:

```
# mdadm /dev/md0 -f /dev/hda8

mdadm: set /dev/hda8 faulty in /dev/md0
```

Mire el estado del RAID en **/proc/mdstat** durante la ejecución:

```
md0: active raid1 hda10[2] hda9[1] hda8[0](F)
    104320 blocks [2/1] [U_]
    [=>..................]  recovery =  8.8% (9216/104320) finish=0.1min
speed=9216K/sec
```

Ha aparecido una "(F)" cerca de hda8, lo que indica un disco Faulty. Vemos también que, de los dos discos, uno tiene una avería y que el RAID reconstruye su matriz con el spare disk. Después de la ejecución, obtenemos:

```
md0: active raid1 hda10[1] hda9[0] hda8[2](F)
     104320 blocks [2/2] [UU]
```

El RAID está reconstruido y funciona de maravilla.

```
# mdadm --detail /dev/md0
...
          State: clean
 Active Devices: 2
Working Devices: 2
 Failed Devices: 1
  Spare Devices: 0
```

```
    Number   Major   Minor   RaidDevice State
       0       3        9        0      active sync   /dev/hda9
       1       3       10        1      active sync   /dev/hda10
       2       3        8       -1      faulty        /dev/hda8
...
```

El disco Faulty es realmente **/dev/hda8**; **/dev/hda10** lo sustituyó como disco de emergencia. Así, el disco de emergencia se convierte en un disco RAID de la matriz.

6. Sustituir un disco

Ya que **/dev/hda8** tiene un problema, lo va a sustituir. Sáquelo con `-r (o --remove)`:

```
# mdadm /dev/md0 -r /dev/hda8
mdadm: hot removed /dev/hda8

# cat /proc/mdstat
Personalities: [raid1]
md0: active raid1 hda10[1] hda9[0]
     104320 blocks [2/2] [UU]
```

Constate que `hda8` ha desaparecido. Puede apagar la máquina y luego sustituir el disco defectuoso. Encienda de nuevo la máquina y vuelva a particionar el disco correctamente. Sólo falta añadir de nuevo el disco arreglado en la matriz RAID con `-a (--add)`:

```
# mdadm /dev/md0 -a /dev/hda8
mdadm: hot added /dev/hda8

# cat /proc/mdstat
Personalities: [raid1]
md0: active raid1 hda8[2] hda10[1] hda9[0]
     104320 blocks [2/2] [UU]
```

El disco `hda8` aparece de nuevo. Vea el detalle:

```
# mdadm --detail /dev/md0
...
         State:  clean
 Active Devices: 2
Working Devices: 3
 Failed Devices: 0
  Spare Devices: 1


    Number   Major   Minor   RaidDevice State
       0       3        9        0      active sync   /dev/hda9
       1       3       10        1      active sync   /dev/hda10
       2       3        8       -1      spare         /dev/hda8
...
```

Se ha añadido de nuevo el disco **/dev/hda8** ¡y el sistema lo ha convertido en el nuevo disco de emergencia!

7. Apagado y puesta en marcha manual

Puede apagar de manera puntual una matriz RAID con `-S (--stop)` DESPUÉS de haber desmontado el periférico:

```
# mdadm --stop /dev/md0
```

La matriz RAID se pone en marcha de nuevo con `-As (--assemble -scan)`. Eso implica que el archivo **`/etc/mdadm.conf`** está correctamente configurado (`--scan` es una opción que busca la información en ese archivo).

```
# mdadm --assemble --scan /dev/md0
```

Si el RAID no arranca de nuevo, puede intentar con `-R (--run)`: es probable que falte un disco o que no se haya acabado aún una reconstrucción en curso:

```
# mdadm --run /dev/md0
```

Introducción al LVM

1. Principio

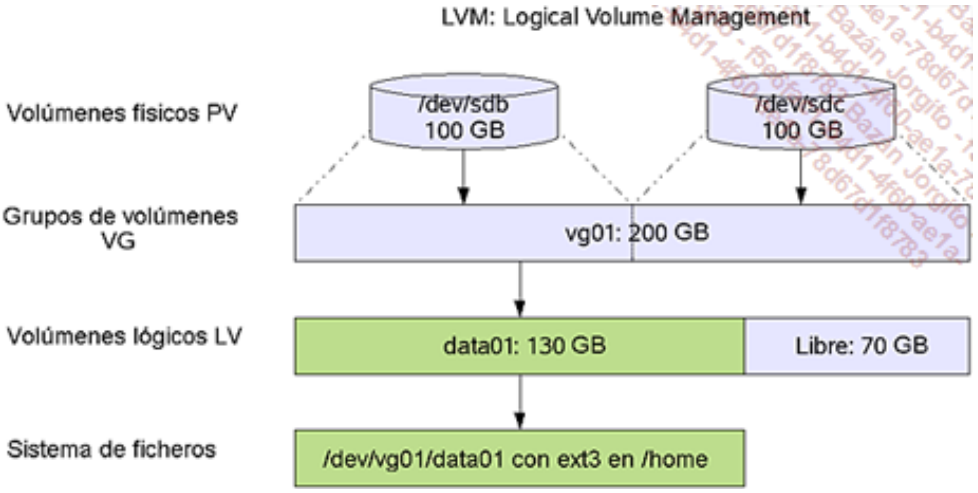
El **Logical Volume Manager** es un sistema de gestión muy perfeccionado de los soportes de almacenamiento. El objetivo es superar, incluso transcender, la gestión física de los discos y su organización lógica básica (las particiones) para extender la capacidad global de los soportes, usando una gestión completamente lógica de ella.

Un LVM permite, del mismo modo que el RAID 0 por ejemplo, crear espacios de datos lógicos en varios discos. Permite también hacer mirroring, como el RAID 1. Pero la comparación no va más allá. El RAID se limita a crear una "partición" en un espacio de almacenamiento definido por el propio RAID (por ejemplo, una partición de 100 GB en un RAID 0 de dos discos de 50 GB).

El LVM agrupa discos físicos o cualquier otro soporte de almacenamiento denominado físico (disco, hardware RAID, software RAID, soporte de almacenamiento procedente de un SAN), que llama a volúmenes físicos PV (*Physical Volume*) en un grupo de volúmenes VG (*Volume Group*). El LVM ve el grupo VG como una especie de metadisco en el cual va a crear volúmenes lógicos LV (*Logical Volume*) a voluntad.

- Volumen físico PV: un soporte físico de almacenamiento de datos: disco duro, por ejemplo;
- Grupo de volúmenes VG: una agrupación lógica de 1 a n VG;
- Volumen lógico LV: un recorte lógico en el interior de un VG.

Se ve el volumen lógico como una partición y se puede utilizar como tal. Puede contener datos; basta con crear un sistema de archivos ordinario (ext3, por ejemplo) y montarlo de manera totalmente clásica.



Al contrario del Software RAID 0, donde la partición de datos debe ocupar todo el espacio, se pueden crear tantos volúmenes lógicos de cualquier tamaño como se desee. Pero va mucho más allá.

El LVM es dinámico. Se pueden añadir y suprimir volúmenes físicos de un grupo de volúmenes. Al añadir volúmenes físicos, la capacidad, y por lo tanto el espacio disponible del grupo, aumenta. El nuevo espacio disponible puede permitir crear nuevos volúmenes lógicos, pero también ampliar un volumen lógico existente.

Un volumen lógico es dinámico: se puede ampliar o reducir a voluntad, lo que implica que también debe poderse ampliar un sistema de archivos o reducirlo.

Apunte también que se puede utilizar una matriz RAID como volumen físico.

Las configuraciones de LVM se encuentran repartidas por los archivos y directorios de **`/etc/lvm`**. El archivo **`/etc/lvm/lvm.conf`** contiene la configuración global. La configuración de los diferentes

volúmenes (físicos, grupos y lógicos) no se encuentra en un archivo, sino en una estructura presente en el interior de los propios periféricos, en sus primeros bloques: son los metadatos de los volúmenes físicos.

2. Los volúmenes físicos

a. Crear un volumen físico

Un volumen físico puede ser un disco completo o una partición clásica en el interior de un disco. En ese caso, la partición debe ser de tipo **0x8e**.

A continuación, tiene la salida del comando **fdisk** en **/dev/sdb**. Distinga entre las particiones primarias 2 y 3 de tipo **8e** que servirán para los ejemplos siguientes.

```
# fdisk -l /dev/sdb

Disco /dev/sdb: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000eab03

Periférico Inicio   Principio       Fin       Bloques    Id  Sistema
/dev/sdb1                   2     16846    135307462+    83  Linux
/dev/sdb2               16847     18152     10490445    8e  Linux LVM
/dev/sdb3               18153     19457     10482412+   8e  Linux LVM
```

Una vez creadas las participaciones, utilice el comando **pvcreate** en una primera partición (se pueden precisar varias particiones):

```
# pvcreate /dev/sdb2
  Physical volume "/dev/sdb2" successfully created
```

b. Ver los volúmenes físicos

El comando **pvdisplay** permite visualizar el conjunto de los volúmenes físicos accesibles en su sistema. También puede llevar el nombre de volumen específico.

```
# pvdisplay /dev/sdb2
  "/dev/sdb2" is a new physical volume of "10,00 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               10,00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

De momento, la información es reducida. El PV no pertenece todavía a ningún grupo de volúmenes (línea `VG Name`). Su tamaño es de 10 GB. Las líneas más interesantes, aunque de momento vacías porque el PV no pertenece a ningún grupo de volúmenes, son aquellas donde está indicado `PE`. PE significa *Physical Extend*, extensión física. Cada VG, y por lo tanto el PV que lo constituye, está dividido en secciones llamadas PE. El PE es la unidad básica de trabajo del LVM.

Si un PE es de 4 MB, significa que se podrá distribuir el espacio en el interior del grupo de volúmenes en secciones de 4 MB. La asignación se hace por PE: por tanto, la creación de un volumen lógico de 500 PE de 4 MB da 2000 MB.

Estos valores pasan a cero en cuanto el PE se integra en un VG.

3. Los grupos de volúmenes

a. Crear un grupo de volúmenes

Para crear un grupo de volúmenes, debe disponer de al menos un volumen físico. Puede crear un grupo de volúmenes con el comando **vgcreate**. Un grupo de volúmenes llevará el nombre que desee. Es el primer argumento del comando. Luego facilite como argumento la lista de los volúmenes físicos que componen el grupo de volúmenes, aquí sólo uno, **/dev/sdb2**.

```
# vgcreate vg01 /dev/sdb2
  Volume group "vg01" successfully created
```

b. Propiedades de un grupo de volúmenes

El grupo de volúmenes dispone de numerosas propiedades. Puede listarlas con el comando**vgdisplay**.

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               10,00 GB
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       0 / 0
  Free  PE / Size       2561 / 10,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

Fíjese en las líneas `MAX LV` y `MAX PV`. La primera indica el número máximo de volúmenes lógicos que se podrá crear en este grupo de volúmenes. El valor cero indica un número infinito en teoría. La segunda indica el número máximo de volúmenes físicos que se pueden añadir al grupo de volúmenes. Aquí también el cero indica un número infinito.

Debe fijarse en el hecho de que Linux representa un caso particular en este campo. El número de LV y PV es aquí virtualmente infinito (como, por ejemplo, el volumen lógico lvm2), delo que no ocurre con los demás UNIX, donde los valores MAX LV, MAX PV y PE Size se determinan automáticamente en el momento de la creación de volúmenes, y en función, entre otras, de las propiedades de los volúmenes físicos que componen el grupo.

Se pueden establecer los valores al crear el grupo de volúmenes usando los siguientes parámetros del comando **pvcreate**:

-l Número máximo de volúmenes lógicos.

-p Número máximo de volúmenes físicos.

-s Tamaño de las extensiones físicas (con un sufijo k, m, g o t para precisar la unidad).

Las últimas líneas se refieren a las PE (extensiones físicas). El grupo dispone actualmente de 2561 extensiones de 4 MB, o sea 10 GB, todas libres. Los volúmenes lógicos ocuparán más tarde un determinado número de estas PE, según su tamaño.

El comando **vgdisplay** acepta el parámetro `-v`, que da más detalles, y en particular la lista de los volúmenes físicos que lo componen.

```
# vgdisplay -v vg01
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               10,00 GB
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       0 / 0
  Free  PE / Size       2561 / 10,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG

  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 2561
```

Ahora que el PV **/dev/sdb2** forma parte de un VG, hay más información disponible:

```
# pvdisplay /dev/sdb2
  --- Physical volume ---
  PV Name               /dev/sdb2
  VG Name               vg01
  PV Size               10,00 GB / not usable 589,00 KB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2561
  Free PE               2561
  Allocated PE          0
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

4. Los volúmenes lógicos

a. Crear un volumen lógico

Un volumen lógico es una sección de un VG (grupo de volúmenes) o, lo que es lo mismo, una partición en la cual podrá crear un sistema de archivos. Un volumen lógico LV ocupa un determinado número de PE (extensiones físicas) de un VG, contiguas o no. Esto tiene su importancia porque:

- se puede ampliar un LV mientras quedan PE libres en el VG.
- se puede reducir un LV, lo que liberará unas PE en el VG, que podrán ser utilizadas para

crear nuevos LV o para ampliarlos.

Esto significa que el LVM gestiona una especie de índice y orden de PE para determinar a qué LV pertenece un PE.

Puede crear un volumen lógico con el comando **lvcreate**. Un volumen lógico lleva un nombre, dispone de un tamaño expresado en extensiones lógicas LE (*Logical Extension*) que corresponden a PE en el interior de un LV, ya sea en KB, MB, GB... El comando siguiente crea un volumen lógico llamado **data01** en el interior del VG **vg01**, de un tamaño de 6 GB. La `-L` precisa que la unidad está en MB (`m`), GB (`g`), TB (Terabyte, `t`), PB (Petabyte) o EB (Exabyte). Para precisar un número de PE, utilice "`-l`".

```
# lvcreate -n data01 -L 6g vg01
  Logical volume "data01" created
```

El sistema muestra un LV como una partición y después de su creación dispone de un archivo periférico asociado. El archivo está en la carpeta **/dev/<nombre_del_vg>/<nombre_del_lv>**. Observe que se trata de un vínculo simbólico hacia un archivo de **/dev/mapper** para mantener la compatibilidad con los demás Unix.

```
# ls -l /dev/vg01/data01
lrwxrwxrwx 1 root root 23 sept. 13 09:27 /dev/vg01/data01 -> /dev/
mapper/vg01-data01
```

b. Propiedades de un volumen lógico

A las propiedades de un volumen lógico se accede con el comando **lvdisplay**:

```
# lvdisplay /dev/vg01/data01
File descriptor 3 left open
File descriptor 4 left open
  --- Logical volume ---
  LV Name                /dev/vg01/data01
  VG Name                vg01
  LV UUID                6ucPwc-sxMJ-K9P3-MkWR-t28I-NyRM-ZKkTmm
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                6,00 GB
  Current LE             1536
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

Puede añadir los parámetros `-v` y `-m`. En este último caso, **lvdisplay** muestra también los segmentos ocupados por el volumen lógico en el interior de los diferentes volúmenes físicos y, por tanto, el reparto de las extensiones físicas ocupadas por el volumen lógico en el interior de cada volumen físico. Como, de momento, sólo hay un PV en el interior del VG, puede obtener lo siguiente:

```
# lvdisplay -m /dev/vg01/data01
...
  --- Segments ---
  Logical extent 0 to 1535:
    Type               linear
    Physical volume    /dev/sdb2
    Physical extents   0 to 1535
```

c. Acceso al volumen lógico

Puede crear un sistema de archivos y montar el LV como cualquier partición:

```
# mkfs -t ext3 /dev/vg01/data01
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Etiqueta de sistema de archivos=
Tipo de sistema operativo: Linux
Tamaño de bloque=4096 (log=2)
Tamaño de fragmento=4096 (log=2)
393216 i-nodos, 1572864 bloques
78643 bloques (5.00%) reservados para el superusuario

Primer bloque de datos=0
Número máximo de bloques del sistema de archivos=1610612736
48 grupos de bloques
32768 bloques por grupo, 32768 fragmentos por grupo
8192 i-nodos por grupo
Superbloques de emergencia almacenados en los bloques:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Escritura de las tablas de i-nodos: completado
Creación del log (32.768 bloques): completado
Escritura de los superbloques y de la información de compatibilidad del
sistema de archivos: completado

Se verificará el sistema de archivos automáticamente cada 37 montajes o
después de 180 días, según la primera eventualidad. Utilizar tune2fs -c o -i
para sobreescribir el valor.
```

Sólo queda montar el nuevo sistema de archivos.

```
# mount -t ext3 /dev/vg01/data01 /mnt/data01
# df /mnt/data01
Sis. de fich.        1K-blocs     Ocupado Disponible Capacidad Montado en
/dev/mapper/vg01-data01
                       6192704    143488    5734644   3%       /mnt/data01
```

5. Ampliaciones y reducciones

a. Los grupos de volúmenes

Hasta aquí todo sigue el guión esperado. La fuerza del LVM reside en su dinamismo. Vayamos ahora a sacar partido de la situación: supongamos que necesita crear un nuevo LV de 6 GB llamado **data02** en el interior del VG **vg01**. A continuación presentamos el estado actual de **vg01**:

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               10,00 GB
```

```
  PE Size               4,00 MB
  Total PE              2561
  Alloc PE / Size       1536 / 6,00 GB
  Free  PE / Size       1025 / 4,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

No hay espacio suficiente, sólo 4 GB (1025 PE). Hay que añadir un nuevo volumen físico en el interior de este VG. Esto se hace con el comando **vgextend**, que funciona de la misma manera que**vgcreate**: indique el nombre del VG seguido del PV o de los PV que quiere añadir.

```
# pvcreate /dev/sdb3
  Physical volume "/dev/sdb3" successfully created
# vgextend vg01 /dev/sdb3
  Volume group "vg01" successfully extended
```

A continuación se muestra el nuevo estado de **vg01**. Observe que el VG contiene ahora dos PV, y que juntos suman un tamaño de 14 GB (3584 PE).

```
# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               20,00 GB
  PE Size               4,00 MB
  Total PE              5120
  Alloc PE / Size       1536 / 6,00 GB
  Free  PE / Size       3584 / 14,00 GB
  VG UUID               dZt8KP-xwol-5mb3-NaVW-Wsui-3sQy-p8kvpG
```

Sólo queda crear el LV **data02** de 6 GB:

```
# lvcreate -n data02 -L 6g vg01
  Logical volume "data02" created
```

Cuando crea un LV, el LVM busca optimizar el uso de las PE, de manera que sean lo más contiguas posible, y si se puede, en un mismo PV. Esto se ve con el comando **lvdisplay** y el parámetro `-m`, en las líneas `Segments` y la lista de los segmentos propuestos.

```
# lvdisplay -m /dev/vg01/data02
File descriptor 3 left open
File descriptor 4 left open
  --- Logical volume ---
  LV Name                /dev/vg01/data02
  VG Name                vg01
  LV UUID                QozsE1-tA70-cj2c-RZeD-HfX0-dTm8-0wlYpj
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                6,00 GB
```

```
  Current LE             1536
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

  --- Segments ---
  Logical extent 0 to 1535:
    Type                linear
    Physical volume     /dev/sdb3
    Physical extents    0 to 1535
```

Los comandos siguientes crean el sistema de archivos en el volumen lógico y lo montan:

```
# mkfs -t ext3 /dev/vg01/data02
# mount -t ext3 /dev/vg01/data02 /mnt/data02
```

b. Ampliar un volumen lógico

Resulta que el LV **data01** de 6 GB es demasiado pequeño. Debe ser el doble. Hay que añadirle 6 GB, lo que es posible porque quedan 8 GB (2048 PE) en el grupo de volúmenes **vg01**:

```
# vgdisplay vg01|grep Free
  Free  PE / Size       2048 / 8,00 GB
```

La ampliación de un volumen lógico se hace en este orden:

- Ampliación del LV con el comando **lvextend**.
- Ampliación del sistema de archivos con **resize2fs** (ext) o **btrfsctl** (btrfs).

Ampliación del LV

El comando **lvextend** autoriza los parámetros `-l` (número de extensiones lógicas LE) o `-L` como para **lvcreate**. A continuación, indique el nuevo tamaño del LV o, si añade un prefijo `+`, el tamaño adicional deseado. También puede indicar, como último argumento, el nombre del PV en el cual forzar la extensión del LV (también es posible con `lvcreate`). Sólo funcionará si el PV o los PV indicados disponen de bastante PE.

El comando siguiente añade 1536 LE (4x1536=6144 MB, o sea, 6 GB) en **data01**:

```
# lvextend -l +1536 /dev/vg01/data01
  Extending logical volume data01 to 12,00 GB
  Logical volume data01 successfully resized
```

Mire ahora en qué PV están ubicados los datos:

```
# lvdisplay -m /dev/vg01/data01
  --- Logical volume ---
  LV Name                /dev/vg01/data01
  VG Name                vg01
  LV UUID                6ucPwc-sxMJ-K9P3-MkWR-t28I-NyRM-ZKkTmm
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                12,00 GB
  Current LE             3072
  Segments               2
  Allocation             inherit
```

```
  Read ahead sectors      auto
  - currently set to      256
  Block device            253:0
  --- Segments ---
  Logical extent 0 to 2560:
    Type                linear
    Physical volume     /dev/sdb2
    Physical extents    0 to 2560

  Logical extent 2561 to 3071:
    Type                linear
    Physical volume     /dev/sdb3
    Physical extents    1536 to 2046
```

El volumen lógico data01 ocupa, en efecto, 12 GB, en dos segmentos de PE, que están en los PV**/dev/sdb2** y **/dev/sdb3**. El LVM asignó por lo tanto un espacio en el conjunto de los PV del VG.

Extensión del sistema de archivos

Sólo el volumen lógico ha sido ampliado. De momento, el tamaño del sistema de archivos contenido en `data01` no ha cambiado:

```
# df -h /mnt/data01
Sis. de fich.          Tam. Oc.   Disp.  %Oc.  Montado en
/dev/mapper/vg01-data01
                       6,0G  141M  5,5G   3%   /mnt/data01
```

El comando **resize2fs** permite reducir y ampliar un sistema de archivos. El primer argumento es el sistema de archivos; el segundo, el tamaño, con un eventual sufijo K (KB), M (MB) o G (GB). Sin sufijo, se indica el número de bloques del sistema de archivos. Si el tamaño está ausente, el sistema de archivos se adaptará al tamaño de la partición o del LV.

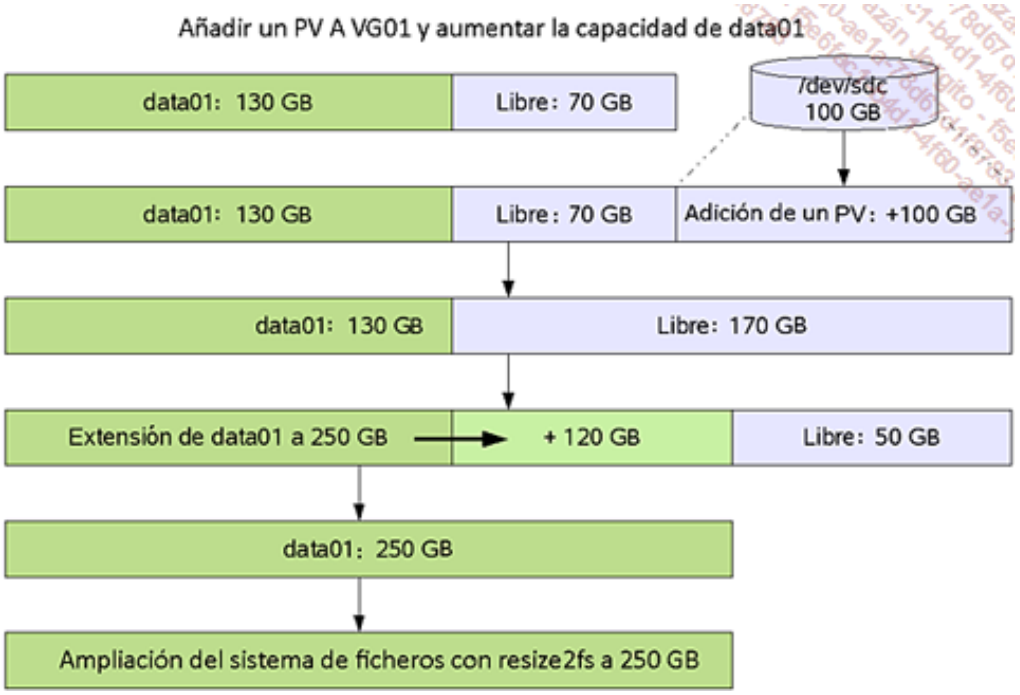
Se puede utilizar el comando **resize2fs** en caliente, es decir, con el sistema de archivos montado, para las ampliaciones. En cambio, habrá que desmontar el sistema de archivos para reducirlo.

```
# resize2fs /dev/vg01/data01
resize2fs 1.40.8 (13-Mar-2008)
Filesystem at /dev/vg01/data01 is mounted on /mnt/data01; on-line
resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/vg01/data01 to 3145728 (4k) blocks.
El sistema de archivos /dev/vg01/data01 tiene ahora un tamaño
de 3145728 bloques.
```

Mire el estado del sistema de archivos: ocupa ahora 12 GB:

```
# df -h /mnt/data01
Sis. de fich.          Tam. Oc.   Disp. %Oc. Montado en
/dev/mapper/vg01-data01
                       12G  144M   12G   2%  /mnt/data01
```

Ahora puede ver la potencia del LVM: adición de nuevos volúmenes físicos y ampliación de volúmenes lógicos en caliente y de manera dinámica. ¿Ya no queda espacio? No importa: basta con añadir un nuevo disco, transformarlo en PV, añadirlo en el VG y volver a dimensionar el LV que no tiene bastante espacio, sin que sea necesario particionar de nuevo, recrear sistemas de archivos, hacer backups, etc.



Un sistema de archivos btrfs también puede aumentar o reducir su tamaño. Al contrario de un sistema de archivos de tipo ext, la reducción se puede hacer en caliente, sin desmontar el sistema de archivos. Para aumentar sus sistema de archivos a la capacidad máxima del LV utilice el comando **btrfsctl**. Observe que se le indica el punto de montaje del sistema de archivos:

```
# btrfsctl -r max /mnt/data01
```

c. Reducir un volumen lógico

Para reducir el tamaño de un volumen lógico, tiene que proceder según el orden siguiente:

- Comprobación del sistema de archivos que hay que reducir con **fsck**, para sistemas de archivos ext.
- Reducción del sistema de archivos contenidos en el volumen lógico con **resize2fs** (ext) o**btrfsctl** (btrfs).
- Reducción del volumen lógico con el comando **lvreduce**.

Va a reducir el LV **data01** a 4 GB. Sólo es posible cuando los datos ocupan menos de 4 GB. En primer lugar, compruebe el tamaño actual del sistema de archivos. En este caso, está casi vacío:

```
# df -h /mnt/data01
Sis. de fich.          Tam. Oc.    Disp. %Oc. Montado en
/dev/mapper/vg01-data01
                       12G  144M  12G    2%   /mnt/data01
```

Sólo se puede reducir el sistema de archivos si no está montado; desmóntelo:

```
# umount /mnt/data01
```

Compruebe el sistema de archivos de tipo ext:

```
# fsck -f /dev/vg01/data01
fsck 1.40.8 (13-Mar-2008)
e2fsck 1.40.8 (13-Mar-2008)Paso 1: verificación de los i-nodos,
de los bloques y de los tamaños
Paso 2: verificación de la estructura de los directorios
```

```
Paso 3: verificación de la conectividad de los directorios
Paso 4: verificación de los contadores de referencia
Paso 5: verificación de la información del sumario del grupo

/dev/vg01/data01: ***** EL SISTEMA DE FICHEROS HA SIDO MODIFICADO *****
/dev/vg01/data01: 11/786432 files (9.1% non-contiguous),
86002/3145728 blocks
```

Vuelva a dimensionar el sistema de archivos a 4 GB:

```
# resize2fs /dev/vg01/data01 4G
resize2fs 1.40.8 (13-Mar-2008)
Resizing the filesystem on /dev/vg01/data01 to 1048576 (4k) blocks.
El sistema de archivos /dev/vg01/data01 tiene ahora un tamaño
de 1048576 bloques.
```

Compruebe el nuevo tamaño del sistema de archivos. 4096*1048576 dan, efectivamente, 4 GB.

```
# dumpe2fs -h /dev/vg01/data01 | grep ^Block
dumpe2fs 1.40.8 (13-Mar-2008)
Block count:              1048576
Block size:               4096
Blocks per group:         32768
```

Para terminar, vuelva a dimensionar el LV a 4 GB. La sintaxis de **lvreduce** es la misma que**lvextend**, excepto que no es posible precisar un PV. Tenga cuidado de no equivocarse aquí: si ha reducido mal el sistema de archivos, puede destruirlo. Conteste **y** a la pregunta sobre si está seguro.

```
# lvreduce -L 4G /dev/vg01/data01
  WARNING: Reducing active logical volume to 4,00 GB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce data01? [y/n]: y
  Reducing logical volume data01 to 4,00 GB
  Logical volume data01 successfully resized
```

Monte de nuevo el sistema de archivos:

```
# mount -t ext3 /dev/vg01/data01 /mnt/data01
# df -h /mnt/data01
Sis. de fich.         Tam.  Oc.  Disp.  %Oc. Montado en
/dev/mapper/vg01-data01
                      4,0G  141M  3,7G   4%  /mnt/data01
```

d. Mover el contenido de un volumen físico

En entornos empresariales es corriente mover un PV hacia otro. Puede ser para sustituir un disco que contiene el PV por otro (para ampliar, por ejemplo). En este caso, puede desplazar el contenido de un PV hacia otro, incluso varios PE de un LV hacia otro PV, o también determinados PE precisos. Sin precisar el destino, el LVM va a mover todos los PE del PV en los otros PV del grupo de volúmenes. Cuidado: los volúmenes físicos deben ser del mismo grupo de volúmenes.

El comando **pvmove** permite mover los PE de un PV hacia otro. Tendrá que desplazar el contenido del PV **/dev/sdb3** a **/dev/sdb2**. **/dev/sdb3** contiene 1536 PE en uso: todos los LE del LV**data02**.

```
# pvdisplay -m /dev/sdb3
  --- Physical volume ---
  PV Name               /dev/sdb3
  VG Name               vg01
  PV Size               10,00 GB / not usable 748,50 KB
```

```
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2559
  Free PE               1023
  Allocated PE          1536
  PV UUID               GwkOvR-DOD0-vpA1-zkVk-1Yb2-gcj3-8HbTl6

  --- Physical Segments ---
  Physical extent 0 to 1535:
    Logical volume      /dev/vg01/data02
    Logical extents     0 to 1535
  Physical extent 1536 to 2558:
    FREE
```

Compruebe si el volumen físico **/dev/sdb2** dispone de bastante espacio para recibir el contenido de **/dev/sdb3**. Quedan 1537 PE en este último, por lo tanto es posible.

```
# pvdisplay /dev/sdb2
  --- Physical volume ---
  PV Name               /dev/sdb2
  VG Name               vg01
  PV Size               10,00 GB / not usable 589,00 KB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              2561
  Free PE               1537
  Allocated PE          1024
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
```

Mueva el PV **/dev/sdb3** hacia el PV **/dev/sdb2**. Puede utilizar el parámetro `-v` para seguir el progreso. Observe que la operación se efectúa cuando no hay ningún sistema de archivos desmontado:

```
# pvmove -v /dev/sdb3 /dev/sdb2
    Wiping cache of LVM-capable devices
    Finding volume group "vg01"
    Found volume group "vg01"
    Found volume group "vg01"
    Checking progress every 15 seconds
  /dev/sdb3: Moved: 4,6%
...
/dev/sdb3: Moved: 97,1%
  /dev/sdb3: Moved: 100,0%
    Found volume group "vg01"
    Found volume group "vg01"
    Loading vg01-data02 table
    Suspending vg01-data02 (253:1) without device flush
    Suspending vg01-pvmove0 (253:2) without device flush
    Found volume group "vg01"
    Found volume group "vg01"
    Found volume group "vg01"
    Resuming vg01-pvmove0 (253:2)
    Found volume group "vg01"
    Resuming vg01-data02 (253:1)
    Found volume group "vg01"
    Found volume group "vg01"
    Removing temporary pvmove LV
    Writing out final volume group after pvmove
    Creating volume group backup "/etc/lvm/backup/vg01" (seqno 9).
```

Ahora compruebe el estado del grupo de volúmenes:

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 1

  PV Name               /dev/sdb3
  PV UUID               GwkOvR-DOD0-vpA1-zkVk-1Yb2-gcj3-8HbT16
  PV Status             allocatable
  Total PE / Free PE    2559 / 2559
```

El segundo PV del **vg01** está totalmente vacío. Por lo tanto, ya se puede suprimir del VG.

e. Reducir un grupo de volúmenes

El comando **vgreduce** permite retirar uno o varios PV de un grupo de volúmenes. Para ello, hace falta primero que los PV en cuestión estén vacíos: sus PE deben estar totalmente libres. Es el caso de **/dev/sdb3**, que va a retirar del VG **vg01**:

```
# vgreduce vg01 /dev/sdb3
  Removed "/dev/sdb3" from volume group "vg01"@
```

Controle que el VG ya no contiene ese PV:

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
    Using volume group(s) on command line
    Finding volume group "vg01"
  --- Physical volumes ---
  PV Name               /dev/sdb2
  PV UUID               KWfJuL-wBmv-ecD1-u1Wt-Ba3d-KK2b-iryDra
  PV Status             allocatable
  Total PE / Free PE    2561 / 1
```

6. Suprimir un grupo de volúmenes

a. Etapas

Para suprimir un grupo de volúmenes, debe seguir las etapas siguientes:

- Desmontar todos los sistemas de archivos de los LV asociados.
- Suprimir todos los volúmenes lógicos con **lvremove**.
- Retirar todos los volúmenes físicos del VG con **lvreduce**.
- Destruir el VG con **vgremove**.

Va a destruir el grupo de volúmenes **vg01**.

b. Suprimir un volumen lógico

Desmonte **data01** y **data02** :

```
# umount /data01
# umount /data02
```

Suprima los volúmenes lógicos con **lvremove**:

```
# lvremove /dev/vg01/data01 /dev/vg01/data02
Do you really want to remove active logical volume "data01"? [y/n]: y
  Logical volume "data01" successfully removed
Do you really want to remove active logical volume "data02"? [y/n]: y
  Logical volume "data02" successfully removed
```

c. Retirar todos los volúmenes físicos

Utilice el comando **vgreduce** con el parámetro `-a` para `ALL`:

```
# vgreduce -a vg01
  Can't remove final physical volume "/dev/sdb2" from volume group
"vg01"
```

Observe que el comando **vgreduce** siempre deja como mínimo un PV en el VG, porque hace falta al menos un PV para constituir un VG.

d. Destruir un grupo de volúmenes

Utilice el comando **vgremove** para destruir un grupo de volúmenes:

```
# vgremove vg01
  Volume group "vg01" successfully removed
```

Compruebe que los archivos y directorios asociados han desaparecido:

```
# ls /dev/vg01
ls: no puede acceder a /dev/vg01: Ningún archivo o carpeta de este tipo
```

Para terminar, el comando **vgdisplay** ya no devuelve nada:

```
# vgdisplay
```

e. Suprimir un volumen físico

Ahora se pueden destruir los dos volúmenes físicos, puesto que ya no están en uso. Puede destruir la información contenida en el volumen con el comando **pvremove**. Sin embargo, si destruye la partición vía `fdisk` o crea un sistema de archivos encima, el efecto es el mismo.

7. Comandos adicionales

Sólo hemos echado un vistazo rápido a las posibilidades del LVM. Tenga presente que cuenta con otros comandos, como por ejemplo:

- **pvchange**: modifica el estado de un volumen físico, por ejemplo para prohibir la asignación de extensiones físicas en este volumen.
- **pvresize**: vuelve a dimensionar un volumen físico si su partición o disco de origen ha sido ampliado o reducido.
- **pvscan**: busca todos los volúmenes presentes en todos los soportes de almacenamiento del sistema.
- **vgchange**: modifica los atributos de un grupo de volúmenes, para activarlo o desactivarlo por ejemplo, pero también para modificar sus valores máximos de PV y de PE, o para prohibir su ampliación o su reducción.

- **vgscan**: busca todos los grupos de volúmenes en todos los soportes.
- **vgrename**: renombra un grupo de volúmenes.
- **vgmerge**: ensambla dos grupos de volúmenes en uno único.
- **lvresize**: vuelve a dimensionar un volumen lógico. Equivale tanto a **lvextend** como a**lvreduce**.
- **lvchange**: modifica los atributos de un volumen lógico.
- **lvrename**: renombra un volumen lógico.

Utilización extendida de BTRFS

1. Los subvolúmenes

a. Un sistema de archivos dentro de otro sistema de archivos

Los subvolúmenes son con frecuencia comparados a los volúmenes lógicos de LVM. Esto es un recurso rápido para explicar una cosa diferente. Un subvolumen BTRFS puede ser visto como un espacio de nombres (namespace) POSIX diferenciado. En la práctica, la creación de un subvolumen se parece a un subsistema de archivos dentro de un sistema de archivos: debe formar parte del sistema de archivos donde ha sido creado, pero es utilizable como si fuera un sistema de archivos totalmente aparte: puede ser montado, disponer de sus propios derechos y cuotas, y puede hacerse una instantánea (un snapshot).

Y sobre todo, al contrario que un volumen lógico, no se crea ningún dispositivo de tipo bloque. Cuando creamos un subvolumen, aparece como una simple carpeta en el entorno de su creación.

La raíz de un sistema de archivos btrfs es en sí un subvolumen, por defecto.

Los subvolumenes se manipulan con el comando **btrfs subvolume**.

b. Creación

Empleamos el comando **btrfs subvolume create**. Aquí un ejemplo de la creación de un subvolumen subvol1 y un ejemplo de su comportamiento:

```
# df -Th .
S. archivos         Tipo    Tamaño    Usados     Disp   Uso%    Montado en
/dev/sdb1           btrfs     5,0G      320K     4,5G     1%    /my_btrfs
# btrfs subvolume create subvol1
      Create subvolume ’./subvol1’
# ls
      subvol1
# df -Th .
S. archivos         Tipo   Tamaño   Usados     Disp   Uso%    Montado en
      -                -      5,0G    352K       4,5G     1%    /my_btrfs/subvol1
```

Podemos constatar dos cosas:

- El subvolumen es considerado como un FS aparte: en el caso inverso las dos primeras columnas indicarían que el dispositivo de tipo bloc /dev/sdb1 y btrfs.
- Sin embargo, al tratarse de un "subsistema de archivos", ocupa el espacio del sistema de archivos inicial y los tamaños idénticos son comunes.

Puede listar los subvolúmenes de esta forma:

```
# btrfs subvolume list /my_btrfs
ID 257 gen 8 top level 5 path subvol1
```

c. Montaje

El principal interés de los subvolúmenes es el de poder montarlos en cadena. También se pueden utilizar opciones de montaje diferentes. Algunas no se admiten y es posible que su lista evolucione según las versiones del núcleo. La opción ro (read only) no se admite hasta la versión del núcleo 3.15. Aquí vemos una demostración. En primera instancia, montamos el subvolumen en /subvol1 mnt, luego escribimos un archivo allí. Posteriormente verificamos que el archivo está en la ubicación inicial.

```
# mkdir /subvol1_mnt
# mount -t btrfs -o subvol=subvol1 /dev/sdb1 /subvol1_mnt/
# df
...
/dev/sdb1              5242880     352    4689856   1% /my_btrfs
/dev/sdb1              5242880     352    4689856   1% /subvol1_mnt
# cd /subvol1_mnt
# touch toto
# ls /my_btrfs/subvol1
toto
# rm /subvol1_mnt/toto
```

Desmontamos un subvolumen como cualquier sistema de archivos:

```
# umount /subvol1_mnt
```

Volvemos a montar el subvolumen para solo lectura e intentamos las mismas operaciones. Constatamos que no podemos escribir en el nuevo punto de montaje, pero que si podemos sobre el subvolumen original:

```
# mount -t btrfs -o subvol=subvol1,ro /dev/sdb1 /subvol1_mnt/
# cd /subvol1_mnt/
# touch titi
touch: no se puede efectuar `touch' sobre « titi »: Sistema de archivos
de sólo lectura
# cd /my_btrfs/subvol1/
# touch titi
# ls titi
titi
```

d. Destrucción

Un subvolumen se destruye de la siguiente forma:

```
# btrfs subvolume delete /my_btrfs/subvol1
Transaction commit: none (default)
Delete subvolume '/my_btrfs/subvol1'
```

Por razones prácticas, el subvolumen ha sido conservado durante el resto del capítulo.

2. Los snapshots

a. Fundamentos

Un snapshot es una instantánea del sistema de archivos en un instante dado. Se trata de captar una "imagen" que represente su estado en el momento de su creación. De esta forma, las modificaciones aportadas a la instantánea o al sistema de archivos son independientes. El tamaño ocupado por una instantánea es el de los datos que se han modificado.

Sólo se puede hacer un snapshot de un subvolumen. Ahora bien, la raíz de un sistema de archivos btrfs es en sí misma un subvolumen, lo que permite hacer un snapshot. Los snapshots son vistos en sí mismos como subvolúmenes.

El interés es evidente: una instantánea permite por ejemplo verificar modificaciones antes de efectuarlas realmente en el sistema de archivos de base. Con btrfs podemos incluso ir más lejos, ya que es posible, si el sistema de archivos raíz es de este formato, hacer una instantánea y utilizarla como raíz: de esta forma puede verificar la actualización de su distribución y si no funciona, volver a la raíz original. Si funciona, puede repartirla en la raíz y reiniciar la actualización, o sincronizar (con

rsync por ejemplo) la instantánea con la verdadera raíz antes de reiniciar.

Las instantáneas tienen otros usos. Con el uso de contenedores (lxc, ocker), podemos presentar una instantánea que se utilizará para trabajar, siempre que se clone todo el contenido: ganar tiempo, ganar espacio. Con la destrucción del contenedor, eliminamos la instantánea: todas las modificaciones posiblemente aportadas desaparecerán. Podemos de esta forma, por ejemplo, crear una instantánea de un entorno de producción en un entorno de prueba, como plantilla, etc. De esta forma nos aseguramos de disponer del último estado de producción para trabajar.

b. Creación

Aquí tenemos el comando que creará una instantánea del subvolumen subvol1 que se montará en /my_btrfs/snap subvol1.

```
# btrfs subvolume snapshot /my_btrfs/subvol1 /my_btrfs/snap_subvol1
Create a snapshot of '/my_btrfs/subvol1' in '/my_btrfs/snap_subvol1'
# cd /my_btrfs/snap_subvol1
# ls
titi
```

Si creamos un archivo en la instantánea, este no aparece en el subvolumen original:

```
# touch loulou
# ls /my_btrfs/subvol1
titi
```

c. Montaje

El montaje funciona exactamente de la misma manera que para los subvolúmenes, con la misma sintaxis de comando.

```
# mkdir /snap_mnt
# mount -t btrfs -o subvol=snap_subvol1, /dev/sdb1 /snap_mnt/
```

d. Destrucción

Un snapshot se suprime exactamente igual que un subvolumen:

```
# btrfs subvolume delete /my_btrfs/snap_subvol1
Transaction commit: none (default)
Delete subvolume '/my_btrfs/snap_subvol1'
```

e. Operaciones con los ID

Puede forzar, empleando el comando **list**, que cada subvolumen disponga de un identificador:

```
# btrfs subvolume list /my_btrfs
ID 257 gen 13 top level 5 path subvol1
ID 259 gen 13 top level 5 path snap_subvol1
```

Partamos de la idea que la raíz del sistema de archivos se encuentra en formato btrfs.

Imagine ahora que desea efectuar operaciones en la raíz, por ejemplo actualizar su distribución, pero que por razones prácticas puede dar marcha atrás en caso de problemas. El ideal es crear una instantánea y trabajar sobre esta. El problema es que desea trabajar sobre la ruta /, y no sobre la ruta de la instantánea.

Antes de nada debemos crear una instantánea de /my_btrfs:

```
# btrfs subvolume snapshot / /snapshot
Create a snapshot of '/' in '/snapshot'
```

Recupere el identificador de la instantánea creada:

```
btrfs subvolume list /snapshot
ID 257 gen 7 top level 5 path snapshot
```

Modifique el id predeterminado del subvolumen por defecto /:

```
# btrfs subvolume set-default 257 /
```

Reinicie. Desplácese a /, verifique que la instantánea (snapshot) no siga apareciendo. Esto es normal ya que / es de hecho la instantánea (snapshot): le ha dado su id.

```
# cd /
# ls
bin  boot  dev  etc  home  lib  lib64  media ...
```

Realice las modificaciones.

Ahora, desea hacer un rollback. Deberá volver al identificador original en /. Como se trata de una raíz del sistema de archivos, el identificador será siempre 0:

```
# btrfs subvolume set-default 0 /
```

Reinicie, y ¡ya !

Si desea aplicar los cambios, se podría montar el id 0 subvolumen en otro punto de montaje, y copiar (a través de rsync) su contenido, y luego poner todo de vuelta en la dirección correcta:

```
# mount -o subvolid=0 /mnt
```

3. Utilizar varios discos

Btrfs permite extender un sistema de archivos a varios discos. Esto es extremadamente simple: si nos falta espacio, añadimos un disco y el espacio estará disponible inmediatamente.

Cuando creamos un sistema de archivos btrfs en varios discos, sin opciones particulares, los metadatos en espejo (copiados idénticamente en cada disco), pero las escrituras son lineales: el primer disco se llena, luego el siguiente, y así sucesivamente. Aquí mostramos como crear un nuevo sistema de archivos en dos discos:

```
# mkfs.btrfs -f /dev/sdb /dev/sdc
Turning ON incompat feature 'extref': increased hardlink limit per file
to 65536
adding device /dev/sdc id 2
fs created label (null) on /dev/sdb
    nodesize 16384 leafsize 16384 sectorsize 4096 size 40.79GiB
Btrfs v3.14.2
```

Este sistema de archivos puede montarse desde cualquier dispositivo de tipo bloc: sdb o sdd.

Btrfs suporta varios modos RAID. RAID5 y RAID6 son todavía experimentales (junio de 2014) y no deben ser usados. RAID 0, 1 y 10 están perfectamente soportados. Aquí vemos como crear un RAID0 (stripping):

```
# mkfs.btrfs -f -d raid0 /dev/sdb /dev/sdc
```

Monte el sistema de archivos y cree en este un archivo de gran tamaño:

```
# mount /dev/sdc /my_btrfs
# cd /my_btrfs
# dd if=/dev/zero of=big_file bs=1024 count=4000000
4000000+0 registros leídos
4000000+0 registros escritos
4096000000 bytes (4,1 GB) copiados, 101,586 s, 40,3 MB/s
# df -H /my_btrfs/
S. archivos 1K-bloques Usados Disp Uso% Montado en
# df -H /my_btrfs/
S. archivos      Tamaño  Usados  Disp Uso%  Montado en
/dev/sdb            44G    4,1G   37G  10%  /my_btrfs
```

El estado del sistema de archivo se obtiene con show:

```
# btrfs filesystem show /mnt
Label: none  uuid: abf5e785-8ced-4216-ab50-65f9c9533e37
    Total devices 2 FS bytes used 3.77GiB
    devid    1 size 20.09GiB used 3.03GiB path /dev/sdb
    devid    2 size 20.71GiB used 3.01GiB path /dev/sdc
```

Añada un disco en caliente de la siguiente forma:

```
# btrfs device add /dev/sdd /my_btrfs
```

Solo hay un pequeño problema: el espacio se añade pero los datos no se extienden al nuevo disco:

```
# btrfs fi show /my_btrfs
Label: none  uuid: 4132375e-9494-41a0-a980-f06bd5556667
    Total devices 3 FS bytes used 3.82GiB
    devid    1 size 20.09GiB used 3.03GiB path /dev/sdb
    devid    2 size 20.71GiB used 3.01GiB path /dev/sdc
    devid    3 size 20.71GiB used 0.00 path /dev/sdd
```

Deberá repartir los datos en el grupo de discos de la siguiente manera:

```
# btrfs balance start -d -m /my_btrfs
Done, had to relocate 7 out of 7 chunks
# btrfs fi show /my_btrfs
Label: none  uuid: 4132375e-9494-41a0-a980-f06bd5556667
    Total devices 3 FS bytes used 3.82GiB
    devid    1 size 20.09GiB used 2.03GiB path /dev/sdb
    devid    2 size 20.71GiB used 3.00GiB path /dev/sdc
    devid    3 size 20.71GiB used 3.03GiB path /dev/sdd
```

Modelo SQL (para el capítulo El shell y los comandos GNU)

Este modelo no es el más óptimo. De hecho, se deberían separar los pedidos en dos tablas: una tabla de pedidos correspondiente a un número único de pedido y toda la información relacionada con ella EXCEPTO los productos (id_pedido, id_cliente, tipo de entrega, posible dirección de entrega si es diferente de la del cliente, fecha del pedido, estado, empleado que lo procesó, fecha de envío, tipo de envío, etc.) y una tabla detalle_pedido relacionando el número de pedido y el número de producto, la cantidad, un posible descuento, etc.

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

-- Estructura de la tabla `t_pedido`

CREATE TABLE IF NOT EXISTS `t_pedido` (
  `id_pedido` int(11) NOT NULL auto_increment,
  `id` int(11) NOT NULL,
  `id_producto` int(11) NOT NULL,
  `ctd` int(11) NOT NULL,
  PRIMARY KEY  (`id_pedido`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Contenido de la tabla `t_pedido`

INSERT INTO `t_pedido` (`id_pedido`, `id`, `id_producto`, `ctd`) VALUES
(1, 1, 3, 1),
(2, 1, 1, 2),
(3, 2, 2, 5);

-- Estructura de la tabla `t_proveedores`

CREATE TABLE IF NOT EXISTS `t_proveedores` (
  `id_proveedor` int(11) NOT NULL auto_increment,
  `apellido` varchar(255) NOT NULL,
  `dirección` varchar(255) NOT NULL,
  `cp` varchar(5) NOT NULL,
  `ciudad` varchar(100) NOT NULL,
  `tel` varchar(10) NOT NULL,
  PRIMARY KEY  (`id_proveedor`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- Contenido de la tabla `t_proveedores`

INSERT INTO `t_proveedores` (`id_proveedor`, `apellido`, `dirección`, `cp`,
`ciudad`, `tel`) VALUES
(1, 'Espacio Multimedia', 'Calle de los pájaros 48', '77120', 'Madrilona',
'160606161'),
(2, 'las verduras asociadas', 'Avenida de los horticultores 14', '28140',
'Lechugas de Madrid', '321414141');

-- Estructura de la tabla `t_productos`

CREATE TABLE IF NOT EXISTS `t_productos` (
  `id_producto` int(11) NOT NULL auto_increment,
  `id_proveedor` int(11) NOT NULL,
  `nombre` varchar(150) NOT NULL,
  `precio` float NOT NULL,
  `iva` float NOT NULL,
  `ctd` int(11) NOT NULL,
  PRIMARY KEY  (`id_producto`),
  KEY `id_proveedor` (`id_proveedor`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Contenido de la tabla `t_productos`
```

```
INSERT INTO `t_productos` (`id_producto`, `id_proveedor`, `nombre`, `precio`,
`iva`, `ctd`) VALUES
(1, 1, 'Llave USB 1 Go', 10.71, 1.196, 20),
(2, 1, 'piña primera calidad', 16.002, 1.196, 30),
(3, 2, 'Rioja Bueno 2009', 14.9835, 1.196, 25);

-- Estructura de la tabla `t_usuarios`

CREATE TABLE IF NOT EXISTS `t_usuarios` (
  `id` int(11) NOT NULL auto_increment,
  `apellido` varchar(100) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `ciudad` varchar(100) NOT NULL,
  `id_alias` int(11) default NULL,
  PRIMARY KEY  (`id`),
  KEY `id_alias` (`id_alias`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Contenido de la tabla `t_usuarios`

INSERT INTO `t_usuarios` (`id`, `apellido`, `nombre`, `ciudad`, `id_alias`)
VALUES
(1, 'García', 'Santiago', 'Madrid', NULL),
(2, 'Aguilera', 'Francisco', 'Alcorcón', 1);
```

LINUX

Dominar la administración del sistema

Este libro sobre **la administración de sistemas Linux** se dirige a todo informático que vaya a gestionar este sistema operativo y que desee aprender o consolidar los conocimientos básicos adquiridos sobre la materia.

Sea cual sea la distribución de Linux empleada (bien sea en entornos empresariales o domésticos), se presentan y detallan todos los métodos y comandos de administración de Linux. Esta nueva edición del libro contempla las **últimas novedades de Linux**.

El libro realiza un recorrido por los conocimientos necesarios para **la instalación** de una distribución, la gestión de **paquetes** de software, la **compilación** a partir de las fuentes, las **librerías** compartidas, los principales comandos **Gnu** y los scripts shell, la gestión de **discos** y **sistemas de archivo**, la implementación de **volúmenes RAID y LVM**, el **arranque y parada** del sistema, **la impresión** y las tareas comunes de administración de la gestión de **usuarios** y **la automatización de tareas**, la configuración de **redes** y los servicios asociados, **el núcleo** y su compilación, las bases de la **seguridad** o la configuración **del entorno gráfico**.

Todos los puntos tratados se complementar con ejemplos y su dominio le convertirá en un administrador de sistemas Linux competente.

Los capítulos del libro

Introducción • Presentación de Linux • Instalación de Linux y de los paquetes de software • El shell y los comandos GNU • Los discos y el sistema de archivos • Inicio de Linux, servicios, núcleo y periféricos • Las tareas administrativas • La red • La seguridad • X Window • Particionamiento avanzado: RAID, LVM y BTRFS • Anexo

Sébastien ROHAUT

Después de pasar 15 años en una empresa de ingeniería informática **Sébastien Rohaut** es hoy en día Ingeniero de Sistemas en un importante grupo Francés. Asimismo, durante más de 10 años dio cursos de preparación de ingenieros, en concreto en el marco de la preparación para los exámenes de Linux. Esta gran experiencia técnica y pedagógica le permite ofrecer a los lectores un libro eficaz sobre la administración de un sistema Linux.

eni ediciones

www.ediciones-eni.com

Para más información:

ISBN : 978-2-7460-9646-2

www.fullengineeringbook.net