

INFORMÁTICA
TÉCNICA



4ª edición

Expert

Linux

Administración del sistema y explotación de los servicios de red



Dogram Code

Philippe BANQUET



¿Qué contiene este libro?

Este libro tiene como objetivo complementar sus conocimientos en administración para por un lado ayudarle a comprender el funcionamiento de GNU/Linux y por otro a optimizar o reparar el sistema. Está destinado a estudiantes de informática, profesionales y a todas las personas que mantienen servidores Linux.

El libro está basado en las tres principales distribuciones que se encuentran como servidores:

- Debian 8
- Ubuntu Server 14.04 LTS
- CentOS 7

El objetivo de este libro es proporcionar una visión general de los diferentes componentes que constituyen una distribución GNU/Linux.

Descubrirá en los capítulos siguientes:

- el kernel de Linux y los módulos del kernel,
- la configuración del sistema dentro de los sub-sistemas de archivo /proc y /sys,
- el soporte hardware, sistema, aplicaciones, red,
- el análisis de rendimiento.

El último capítulo se refiere a los puntos clave a tener en cuenta para securizar un servidor Linux.

Requisitos previos

Para abordar cómodamente la lectura del presente libro, es indispensable tener buenas nociones acerca de:

- el hardware informático,
- la administración de un sistema GNU/Linux, en concreto acerca de las distribuciones Debian, Ubuntu Server y CentOS,
- el shell bash y la programación de scripts,
- el funcionamiento de las redes locales,
- el protocolo de comunicaciones TCP/IP,
- el lenguaje C.

Obtener el software a implementar

Para reproducir las diferentes operaciones descritas en el libro, podemos instalar las distribuciones GNU/Linux mencionadas en máquinas virtuales. Para ello, debemos obtener el software necesario.

1. Imagen ISO de las distribuciones Linux

En el siguiente cuadro, encontrará las direcciones de los sitios web para descargar las imágenes ISO de las distribuciones Linux.

Distribución	Dirección
Debian 8	https://www.debian.org/CD/http-ftp/index.es.html#stable
Ubuntu Server 14.04 LTS	http://www.ubuntu.com/download/server
CentOS 7	https://www.centos.org/download/

2. Oracle VM VirtualBox

El software que nos permitirá la virtualización de los equipos es Oracle VM VirtualBox versión 4 o 5 (licencia GPL versión 2) con su extensión Oracle VM VirtualBox Extension Pack (licencia PUEL o Personal Use and Evaluation License). Estos programas funcionan con Microsoft Windows, GNU/Linux y OS X, tanto en 32 bits como en 64 bits.

Su equipo deberá contar con la siguiente configuración:

- Un procesador reciente Intel o AMD (por lo menos un Intel Core 2 Duo).
- Debemos identificar la memoria RAM necesaria para el sistema anfitrión (host operating system) con la que contamos. Las máquinas virtuales que vamos a emplear ocupan 512 MB cada una.
- El disco empleado por las máquinas virtuales no excederá los 30 GB. Bastará con añadir 30 MB aprox. por VirtualBox más el tamaño del sistema anfitrión.

Oracle VM VirtualBox y su Pack de extensión están disponibles en la siguiente dirección: <https://www.virtualbox.org/wiki/Downloads>

3. Máquinas virtuales

Crearemos tres máquinas virtuales siguiendo casi la misma filosofía.

Características de la máquina virtual Debian 8

Parámetros	Opciones	Valores
General - Simple	Nombre	Debian
	Tipo	Linux
	Versión	Debian (64 bits)
Sistema - Placa base	Memoria RAM	512 MiB
	Orden de arranque	Disco duro Disco CD/DVD Desactivar disquetera
Sistema - Procesador	Número de procesadores	2
Almacenamiento	Controladora IDE	Asignar la imagen ISO de Debian 64 bits.
	Controladora SATA	Crear un disco de tamaño dinámico de 50 GiB. Crear cuatro discos de tamaño dinámico de 10 GiB respectivamente.
Red	Tarjeta 1	NAT

¿Qué tipo de instalación?

Durante la instalación, seleccione **Advanced options** y luego **Graphical expert install**.

Gestión de cuentas

Durante el proceso de instalación, desactive la cuenta root. Cree un usuario:

- nombre: bob
- contraseña: pa\$\$w0rd

¿Qué tipo de particionado?

Cree en el primer disco de 50 GB un grupo de volúmenes: systemVG. En él crearemos cuatro volúmenes lógicos:

Nombre de Volumen Lógico	Punto de montaje	Tamaño
rootLV	/	15 GB
swapLV	- -	512 MB
optLV	/opt	170 MB
homeLV	/home	El resto

Características de la máquina virtual Ubuntu Server 14.04 LTS

Parámetros	Opciones	Valores
General - Simple	Nombre	Ubuntu
	Tipo	Linux
	Versión	Ubuntu (64 bits)
Sistema - Placa base	Memoria RAM	512 MB
	Orden de arranque	Disco duro Disco CD/DVD Desactivar disquetera
Sistema - Procesador	Número de procesadores	1
Almacenamiento	Controladora IDE	Asignar imagen ISO de Ubuntu Server.
	Controladora SATA	Crear un disco de tamaño dinámico de 20 GB. Crear cuatro discos de tamaño dinámico de 10 GB respectivamente.
Red	Tarjeta 1	NAT

Gestión de cuentas

Durante el proceso de instalación, cree un usuario:

- nombre: bob
- contraseña: pa\$\$w0rd

¿Qué tipo de particionado?

Crearemos en el primer disco de 50 GB un grupo de volúmenes: systemVG. En él crearemos dos volúmenes lógicos:

Nombre de Volumen Lógico	Punto de montaje	Tamaño
rootLV	/	19,24 GB
swapLV	- -	508 MB

Características de la máquina virtual CentOS 7

Parámetros	Opciones	Valores
General - Simple	Nombre	CentOS
	Tipo	Linux
	Versión	Red Hat (64 bits)
Sistema - Placa base	Memoria RAM	512 MB
	Orden de arranque	Disco duro Disco CD/DVD Desactivar disquetera
Sistema - Procesador	Número de procesadores	2
Almacenamiento	Controladora IDE	Asignar la imagen ISO de CentOS 6.4 64 bits.
	Controladora SATA	Crear un disco de tamaño dinámico de 20 GB. Crear cuatro discos de tamaño dinámico de 10 GB respectivamente.
Red	Tarjeta 1	NAT

Gestión de cuentas

Durante el proceso de instalación, la cuenta root se encuentra activada:

- nombre: root
- contraseña: pa\$\$w0rd

¿Qué tipo de particionado?

Cree en el primer disco de 50 GB un grupo de volúmenes: systemVG. Aquí creamos dos volúmenes lógicos:

Nombre de Volumen Lógico	Punto de montaje	Tamaño
rootLV	/	18,54 GB
swapLV	--	992 MB

4. Convenciones

A lo largo del libro, encontrará líneas de comando en un formato específico:

```
# cat /home/bob/.profile
# Fichier .profile
# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```


Si el primer carácter de la primera línea es un \$, la conexión se ha realizado con una cuenta ordinaria:

```
$
```

Si se encuentra con un #, la conexión se ha efectuado con la cuenta de súper usuario, o sea root:

```
#
```

Para conectarse como root, el método difiere en función de la distribución.

Las demás líneas que comienzan con un # son comentarios.

Debian y Ubuntu

La cuenta root está bloqueada durante la instalación. La cuenta de usuario creada durante la instalación es ahora un sudoer.

Ejemplo

Ahora debe conectarse como el usuario bob:

```
$ logname  
bob
```

Luego efectúe un sudo -i:

```
$ sudo -i  
[sudo] password for bob:
```

Escriba la contraseña de bob y el símbolo del sistema cambiará:

```
$ sudo -i  
[sudo] password for bob:  
#
```

Para verificar, escriba:

```
# logname ; whoami  
bob  
root
```

El comando logname devuelve el nombre de la conexión principal, es decir bob, mientras que whoami devuelve la conexión actual.

Para salir de la sesión root, teclee exit.

CentOS

Puede emplear una conexión principal como el usuario root. Sin embargo, si su conexión principal es bob, entonces podrá utilizar el comando su.

Ejemplo

Abra una sesión principal con la cuenta bob creada durante la instalación. El símbolo de sistema de su terminal es un \$.

Emplee su para cambiar bob por root:

```
$ su -  
Contraseña
```

Introduzca la contraseña de root:

```
$ su -  
Contraseña  
#
```

Para salir de la sesión root, teclee exit.

Una novedad acerca de CentOS 7: el grupo wheel se encuentra activado por defecto (al contrario que en versiones anteriores) en el archivo /etc/sudoers.

```
# grep ^ %wheel /etc/sudoers  
%wheel  ALL=(ALL)        ALL
```

Puede crear un usuario normal bob y luego agregarlo al grupo wheel. De esta forma, esta cuenta se convierte en sudoer.

```
# useradd bob && usermod -G wheel bob && id bob  
uid=1000(bob) gid=1000(bob) grupos=1000(bob),10(wheel)
```

Sinopsis

Es importante conocer la arquitectura y el funcionamiento del sistema operativo para garantizar la instalación, configuración, explotación y mantenimiento.

Este capítulo nos permitirá estudiar las diferentes capas que constituyen el sistema GNU/Linux, proporcionar una definición a algunos términos y describir sus roles específicos.

1. Proyecto GNU

En 1984, Richard Stallman inició el proyecto GNU financiado por la FSF (Free Software Foundation) para crear un sistema libre/gratis de tipo Unix que llamó GNU (GNU's Not Unix). Incluía el compilador GCC (GNU Compiler Collection), GNU Emacs, el shell Bash (Bourne Again Shell), librerías, etc.

El kernel oficial Hurd encontró algunas dificultades durante su desarrollo para elaborar un sistema completo GNU/Hurd. A principios de los años 90, apareció el núcleo (kernel) Linux. El sistema GNU integró a este último y así vio la luz GNU/Linux en la plataforma x86.

2. Distribución GNU/Linux

Las distribuciones GNU/Linux como Debian, Ubuntu, Red Hat, CentOS son sistemas operativos basados en el kernel de Linux con las herramientas GNU. Todas están diseñadas de acuerdo con sus objetivos y su propia filosofía.

GNU/Linux es un sistema operativo modular. Es decir, que incluye un conjunto de proyectos, desarrollados por diferentes equipos, que deben interactuar entre sí: un software de arranque (boot loader), un núcleo (kernel), los controladores de dispositivos (device drivers), un intérprete de comandos (shell), un software de instalación, herramientas de configuración, herramientas de administración del sistema, un cortafuegos (firewall), programas ofimáticos, software de servidores (web, correo, base de datos...), las herramientas de programación (gcc), etc. La lista es larga y varía según la distribución.

El comando uname muestra el tipo de sistema.

Sintaxis

```
uname <option>
```

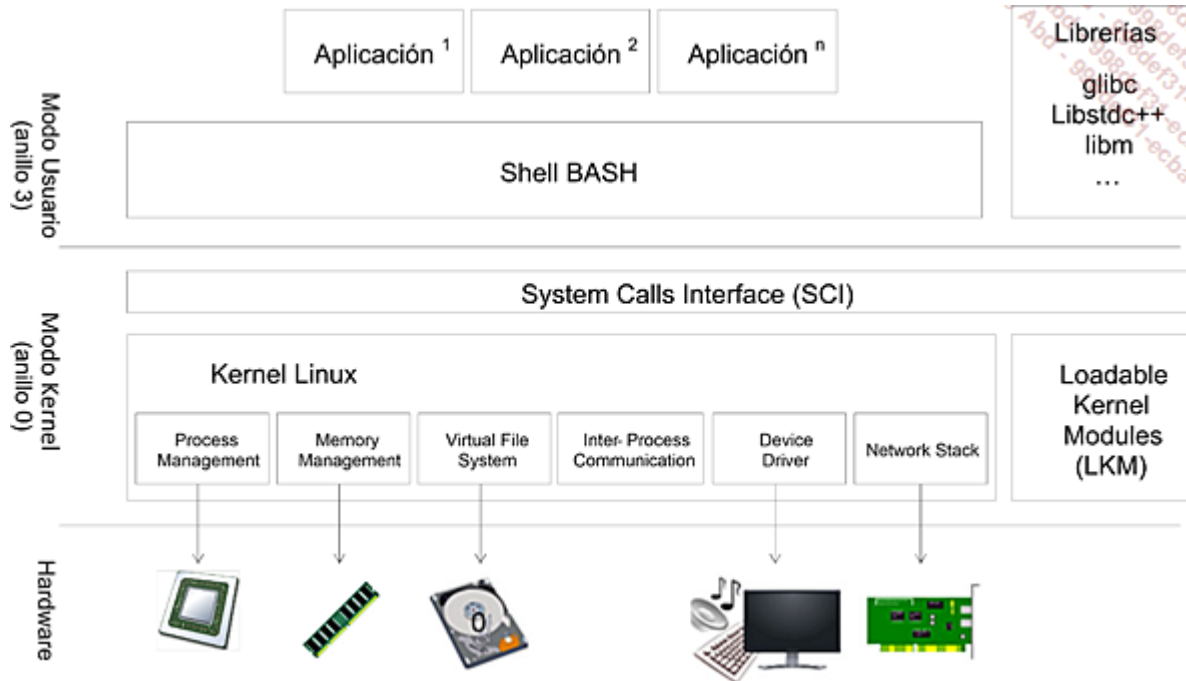
Opción del comando:

Opción		Descripción	Escriba en un terminal para verificar si el sistema
corta	larga		
-o	--operating-system	Mostrar el tipo de sistema.	

operativo es GNU/Linux:

```
# uname -o  
GNU/Linux
```

El esquema siguiente presenta las diferentes capas que constituyen el sistema:



Podemos constatar que, entre varias distribuciones, los componentes pueden ser diferentes o por el contrario comunes, como el kernel o el shell por ejemplo. Dicho esto, la versión o las opciones de compilación de estos últimos pueden variar.

Vamos a estudiar desde la capa hardware hasta la de aplicación, es decir de abajo a arriba.

Anillos de protección

Los sistemas operativos de hoy en día ofrecen por lo general anillos de protección (protection rings) que proporcionan diferentes niveles de privilegios: de menor a mayor. Estos últimos los crea el procesador y no el sistema operativo.

Cuando un programa se ejecuta el sistema operativo le asigna una zona de memoria RAM que se asignan a un anillo de protección. El objeto es separar los programas sensibles, como el kernel del sistema, y los programas de o de los usuarios.

Un programa contenido en una zona de memoria asignada al anillo 3 no podrá acceder a una zona de memoria asignada al anillo 0.

La organización de los anillos de protección varía en función de si tenemos una instalación física o virtual. Y por otra parte, si la arquitectura es de 32 bits o 64 bits.

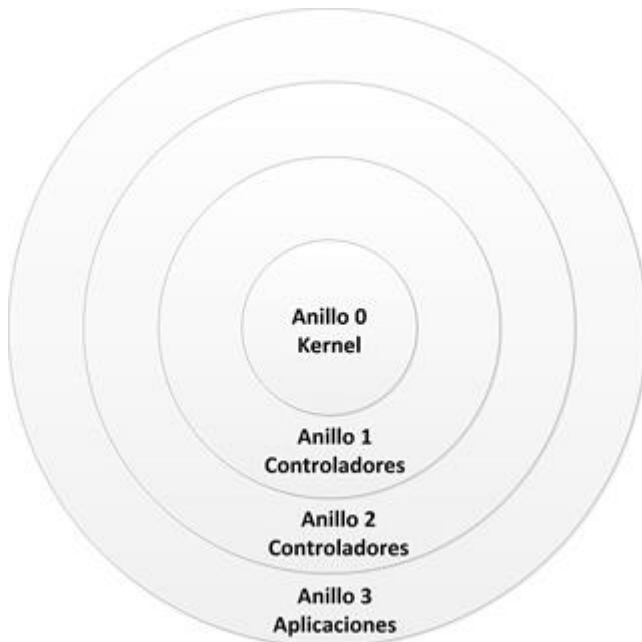
1. Arquitectura de 32 bits

La arquitectura x86 incluye cuatro anillos de protección (rings) numerados del 0 al 3:

- Anillo 0 (ring 0) es el nivel de privilegio más alto, llamado modo supervisor o modo kernel.
- Anillos 1 y 2 (rings 1 y 2) se utilizan para los controladores de dispositivos.
- Anillo 3 (ring 3) es el nivel de privilegio más bajo, llamado modo usuario.

a. Sin virtualización

Por motivos de rendimiento y a su vez de portabilidad a otras plataformas, GNU/Linux 32 bits utiliza el anillo 0 para el espacio del kernel y el anillo 3 para el espacio de usuario. Por lo tanto, los anillos 1 y 2 no se utilizan.

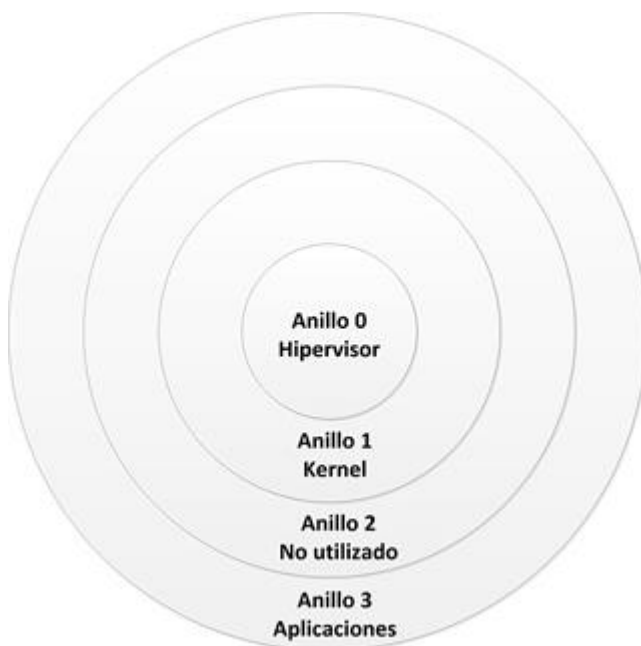


b. Con virtualización

Dentro del marco de la paravirtualización (conocido también con el nombre de hipervisor de Tipo 1), el hipervisor se encuentra en el anillo 0.

Como recordará, el Anillo 0, llamado también en la arquitectura x86 modo supervisor, suele ser el lugar donde se ejecuta el kernel del sistema operativo. Podemos considerar que un kernel es un supervisor.

Un hipervisor gestiona todos los sistemas virtualizados y sirve de interfaz con los recursos físicos. Se sitúa en el anillo 0. Por razones de seguridad, el kernel del sistema virtualizado se ubicará en el anillo 1. En cuanto a las aplicaciones, estas se mantienen en el anillo 3. Seguimos sin utilizar el anillo 2.

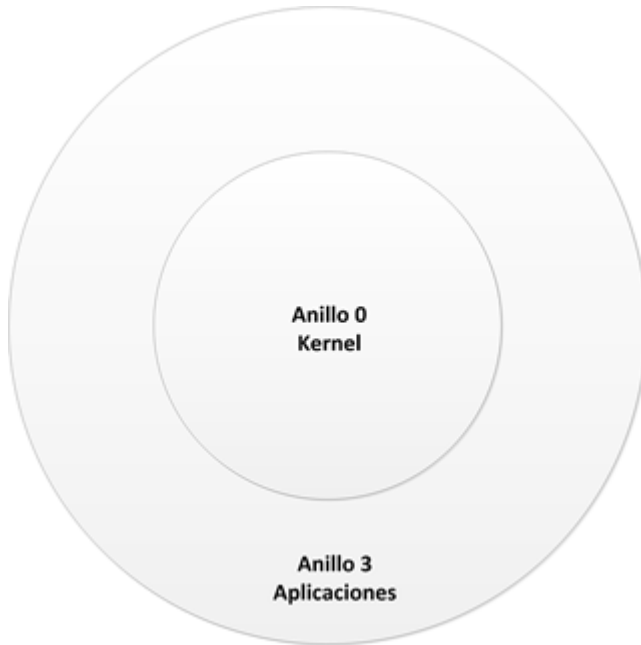


2. Arquitectura de 64 bits

Los niveles de protección evolucionaron con la arquitectura de 64 bits.

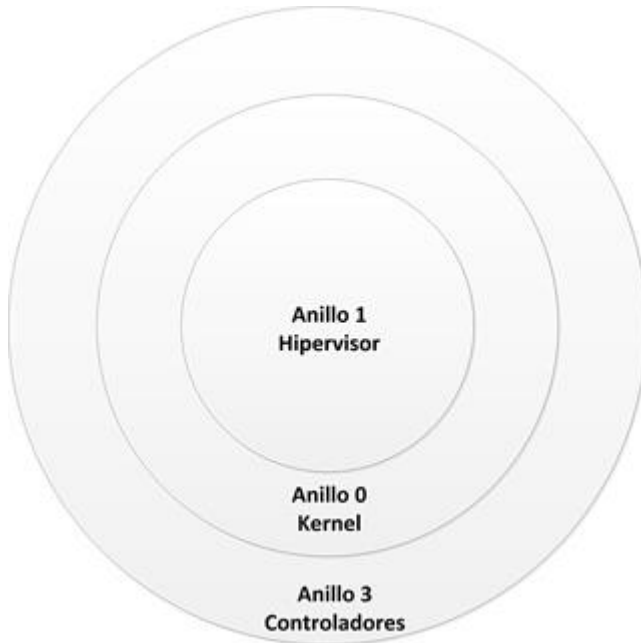
a. Sin virtualización

La arquitectura x86_64 o AMD64 proporciona solo dos anillos de protección que corresponden al anillo 0 para el espacio del núcleo y el anillo 3 para el espacio de las aplicaciones, al igual que en la arquitectura x86.



b. Con virtualización

Debido a la importancia de la virtualización, Intel y AMD mejoraron sus microprocesadores añadiendo un tercer anillo. El anillo -1 contiene al hipervisor. En cuanto a los anillos 0 y 3, no existe ningún cambio.



Plataformas hardware

1. ¿En qué plataformas hardware?

El kernel de Linux se ha portado a una lista exhaustiva de plataformas hardware de 32 bits y de 64 bits. Dec ALPHA, MIPS, X86, x86_64, IA-64, PowerPC, ARM, SPARC, etc.

Encontrará Linux en súper-ordenadores, smartphones, tabletas, servidores, sistemas embebidos (televisión, modem, GPS...) y también, aunque con menos frecuencia, en estaciones de trabajo...

2. Identificador de la arquitectura hardware

El comando `uname`, que ya hemos utilizado, ofrece tres opciones:

Opción		Descripción
corta	larga	
-m	--machine	Muestra la arquitectura de la máquina.
-p	--processor	Muestra el tipo de procesador.
-i	--hardware-platform	Muestra la plataforma hardware.

Para conocer su arquitectura de hardware, introduzca en un terminal `uname` seguido del parámetro `-i`:

```
# uname -i
x86_64
```

Los parámetros `-p` y `-m` son también válidos.

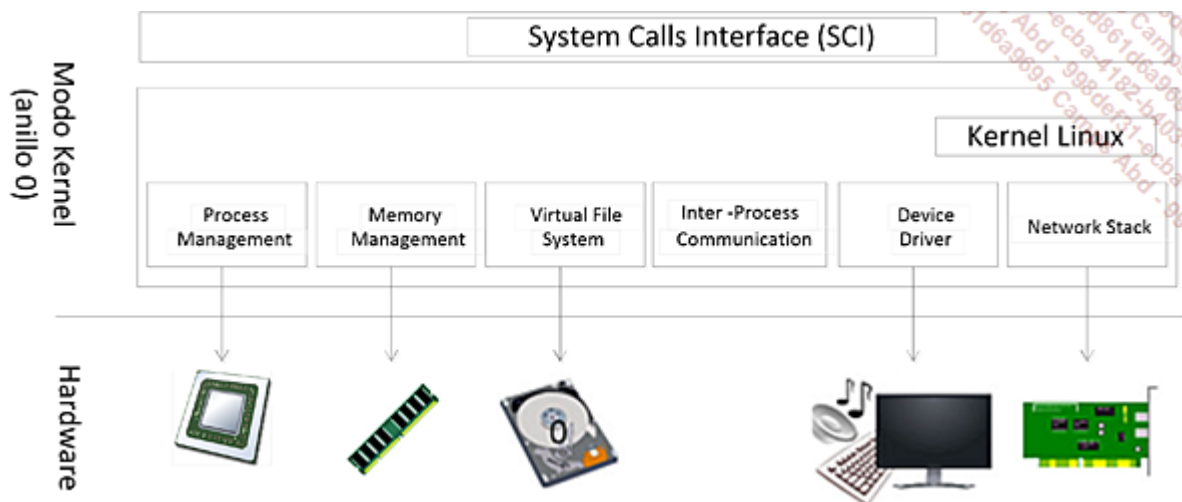
Ubuntu y CentOS devuelven valores con las tres opciones. Sin embargo, Debian muestra «unknown» para las opciones `-i` y `-p`. En este caso podemos utilizar la opción `-m`. Si el equipo es un `x86_64`, el procesador lo será también.

Kernel Linux

Un núcleo (kernel) es un componente importante dentro del sistema operativo que gestiona los recursos del ordenador y sirve de interfaz entre el hardware y los programas.

1. Componentes del kernel

El kernel Linux se ejecuta en el anillo de protección 0:



El kernel de Linux está compuesto por seis subsistemas principales:

- El Process Management (PM) que se encarga de repartir de forma equitativa el acceso al procesador entre todas las aplicaciones activas.
- El Memory Management (MM) que se encarga de asignar a cada programa una zona de memoria que no debe ser leída o modificada por otro proceso.
- El Virtual File System (VFS) que garantiza una gestión correcta de los archivos y un control de los permisos de acceso (ACL - Access Control List). Para reducir la complejidad asociada a los múltiples sistemas de archivos existentes, el kernel emplea llamadas de sistema idénticas e independientes del sistema de archivos escogido. El kernel de Linux encamina los comandos o llamadas estándar a las llamadas específicas del sistema de archivos.
- El Inter-Process Communication (IPC) permite a las aplicaciones comunicarse entre sí de forma que un proceso solo pueda acceder a la zona de memoria que tiene asignada.
- El Device Driver (DD) administra los recursos hardware con los controladores de dispositivos (device drivers) y proporciona a los programas una interfaz uniforme para el acceso a estos recursos.
- El Network Stack (NET) permite conectarse a otros sistemas a través de una red informática. Se soportan muchos dispositivos hardware. Pueden utilizarse varios protocolos de red como IPX/SPX, TCP/IP versión 4 o 6...

2. Código fuente

El kernel de Linux, creado en 1991 por Linus Torvalds, se distribuye bajo los términos de la GNU General Public License GNU (GPL) versión 2. Existen dos ramas de desarrollo mainline y vanilla. Esta última suele ser utilizada por las distribuciones. La descarga del código fuente se realiza desde el sitio web <http://www.kernel.org>.

El código fuente está escrito en su mayoría en lenguaje C como todos los sistemas de la familia UNIX (UNIX Like). El código de bajo nivel está escrito en lenguaje ensamblador.

Se compila con gcc (The GNU Compiler Collection). Cabe señalar que no todo compilador ANSI C podrá necesariamente compilar el kernel ya que GNU C tiene sus propias extensiones que no figuran en el lenguaje C estándar definido por la ISO.

Veremos en el capítulo Kernel Linux cómo compilar, instalar o desinstalar un kernel.

3. Versión del kernel

a. Breve historia de las versiones

La difusión de la primera versión 0.01 del kernel de Linux es muy confidencial en septiembre de 1991 y contiene aproximadamente 10.000 líneas de código.

La versión 0.02, anunciada en la Usenet en octubre de 1991, es difícilmente utilizable como tal. Varias versiones le seguirán para aportar nuevas funcionalidades.

En marzo de 1994, el kernel de Linux versión 1.0.0 se encuentra estable para el uso en producción. Una año más tarde, aparece la versión 1.2.0 para incluso soportar varias arquitecturas de procesadores.

La versión 2.0.0 aporta, en julio de 1996, de forma principal el soporte de equipos de varios procesadores, el soporte de nuevo hardware y una mejor gestión de la red. Luego, evoluciona a las versiones 2.2.0, 2.4.0 y 2.6.0. La versión 2.6.30 cuenta con aproximadamente 11,5 millones de líneas de código.

Linus Torvalds decide en 2011, para los veinte años de Linux, la aparición del kernel 3.0.0 en lugar del 2.6.40. Se continua evolucionando con las versiones 3.1.0, 3.2.0, etc.

La versión estable del kernel 4.0 se anuncia el 12 de abril de 2015. He aquí algunas novedades:

- Los portátiles Toshiba están mejor gestionados por el controlador `toshiba_acpi`.
- El livepatch, que consiste en actualizar el kernel sobre la marcha, es decir, sin reiniciar el equipo.
- Se soporta la gestión de los paneles táctiles FocalTech.
- El soporte para las tarjetas inteligentes Trusted Platform Module 2.0.

El kernel evoluciona a la versión 4.1 en junio de 2015 y aporta de forma principal:

- El soporte ACPI para arquitecturas ARMv8 (64 bits).
- La integración de los sistemas monotarjeta de Qualcomm MSM8916 y Xilinx ZynqMP.
- La implementación de la RFC 7217 que permite almacenar una dirección IPv6 anónima. La dirección MAC no se publica en la dirección IP.
- Muchos cambios introducidos para la plataforma x86 en cuanto a llamadas del sistema, interrupciones, etc.

La versión 4.2 del kernel apareció el 30 de agosto de 2015. Sus principales novedades son las siguientes:

- El soporte para el nuevo procesador Renesas-H8-300 (Versión 32 bits), de NVIDIA Tegra HDA, así como las siguientes nuevas plataformas Broadcom: Buffalo WXR-1900DHP, SmartRG SR400ac y ASUS RT-AC87U, etc.
- Un nuevo algoritmo de gestión de congestión para el protocolo TCP que permite evitar que una sola conexión abarque la totalidad del ancho de banda.
- ext3 está en proceso de desaparición.

La versión 4.3 se publicó en noviembre de 2015. Los principales cambios aportan:

- Muchos cambios en los controladores de dispositivos, en particular gráficos, como Nvidia o la R9 Fury de AMD.
- El soporte de los procesadores Skylake de Intel, ARMv8.1.
- Correcciones para los sistemas de archivos btrfs, F2fs ext4, XFS. Cabe destacar la desaparición de ext3.

La versión 4.4, publicada en enero de 2016, aporta principalmente:

- Una gestión más fina de la memoria, lo que permitirá reducir determinados consumos de memoria excesivos.
- Mejoras en la gestión de los procesadores ARM.
- La herramienta perf, que permite rastrear el kernel, cuenta con nuevas funcionalidades.
- Correcciones de los controladores gráficos AMD, Broadcom, Intel y Nvidia.
- La evolución de los protocolos IPv4 e IPv6.

b. Números de versión

La versión del kernel está compuesta, como hemos visto antes, por varios números.

- El primero es el número mayor.
- El segundo es el número menor. Si su valor era impar, indicaba una versión de desarrollo, en caso contrario el valor par indicaba una versión estable. Este concepto ya no es cierto a partir del kernel 2.6.
- El tercer número de revisión indica que se introducen correcciones de errores («bugs»), de seguridad o nuevas funcionalidades. Por ejemplo: 2.6.39.
- El cuarto número de "patch level" indica la integración de parches de corrección de errores (bugs), de seguridad o de optimización sin añadir nuevas funcionalidades. Por ejemplo: 3.5.0-28.

Más allá de los cuatro números de la versión, contamos con información específica sobre la distribución.

c. ¿Cómo saber la versión del kernel?

Las distribuciones ofrecen una versión del kernel estable y fiable para producción. Por lo tanto, no se precipitan a incorporar a cualquier precio la última versión.

Para identificar la versión utilizada, hablaremos de dos posibilidades:

- Leer el contenido del archivo `/proc/version`.
- Usar el comando `uname` con las siguientes opciones:

Opción		Descripción	La operación es idéntica en las distribuciones que abordamos en este libro. De modo que vamos a efectuar un solo ejemplo.
corta	larga		
-s	--kernel-name	Muestra el nombre del kernel.	Ejemplos con Debian Opción 1: <pre># cat /proc/version Linux version 3.16</pre>
-r	--kernel-release	Muestra el número de versión del kernel.	
-v	--kernel-version	Muestra la versión del kernel.	

```
.0-4-amd64 (debian-kernel@lists.debian.org)
(gcc version 4.8.4 (Debian 4.8.4-1) )
#1 SMP Debian 3.16.7-ckt11-1+deb8u3 (2015-08-04)
```

Opción 2:

```
# uname -s
Linux
# uname -r
3.16.0-4-amd64
# uname -v
#1 SMP Debian 3.16.7-ckt11-1+deb8u3 (2015-08-04)
```

En el cuadro siguiente, tenemos la versión del kernel utilizado en cada distribución.

Distribución	Versión del kernel
Debian 8	3.16.0-4-amd64

Ubuntu Server 14.04 LTS	3.13.0-48-generic
CentOS 7	3.10.0-29.14.1.el7.X86_64

Debian muestra amd64 mientras que CentOS muestra x86_64. Estos valores son similares. Esto significa que el kernel está en versión 64 bits para las arquitecturas Intel y AMD.

Antes de la versión 12.04, Ubuntu ofrecía un kernel optimizado para la versión servidor (server) y otro kernel optimizado para los puestos de trabajo (generic). Ahora, los paquetes linux-image-server y linux-image-generic se han fusionado. Ahora no tenemos más que un solo kernel "generic".

Existen otras versiones:

- El kernel de 32 bits de linux-generic-pae (Physical Address Extension) permite el direccionamiento de memoria más allá del límite de 3,2 GB hasta 64 GB.
- El kernel de linux-rt (real time) está diseñado para una utilización en tiempo real.
- El kernel de linux-virtual integra los controladores necesarios para ser un sistema invitado en KVM, Xen, etc.

4. Módulos del kernel

A partir de la versión 2.0, el kernel de Linux es modular, lo que significa que puede aumentar sus capacidades mediante el uso de módulos (Loadable Kernel Modules o abreviado LKM) como controladores de dispositivos, un cortafuegos (firewall), protocolos de red, etc. Estos últimos pueden cargarse o descargarse dinámicamente sin necesidad de recompilar el kernel.

El hecho de crear una funcionalidad en forma de módulo en lugar de integrarla dentro del núcleo evita sobrecargar el kernel. A mayor tamaño del kernel, mayor riesgo de que su rendimiento se vea afectado. No es necesario incluir en el núcleo los controladores que no se emplean a menudo o nunca.

El código fuente no se suministra con el del kernel. El equipo de desarrollo de una distribución GNU/Linux aporta un conjunto de módulos y nada le impide añadir un controlador adicional.

Para el uso de los módulos, sin embargo, el kernel de Linux debe ser compilado con la opción que permite gestionar los módulos. Las distribuciones ofrecen generalmente un kernel con esta opción activada.

El comando lsmod muestra los módulos cargados:

<https://dogramcode.com/libros-sistemas>

```
# lsmod
Module                Size  Used by
...
nfsd                   263032  2
auth_rpcgss            51211  1 nfsd
oid_registry           12419  1 auth_rpcgss
nfs_acl                12511  1 nfsd
nfs                    188136  0
lockd                  83389  2 nfs,nfsd
*** Salida recortada ***
```

Estudiaremos con más detalle el manejo de los módulos en el capítulo Módulos.

Controladores de dispositivos

Un controlador de dispositivo (device driver o simplemente driver) es un programa que permite a los demás programas del sistema comunicarse con un dispositivo.

GNU/Linux dispone de tres categorías de dispositivos.

- En modo carácter (char drivers). Se comunican con el dispositivo empleando un flujo de bytes a través de puertos series y paralelos o un terminal.
- En modo bloque (block drivers). Se comunican usando bloques de datos con el dispositivo como por ejemplo un disco duro.
- Red (net). Están destinados a controlar los recursos de red.

1. Archivos especiales

Los controladores en modo carácter o en modo bloque están accesibles a través de archivos especiales llamados nodos (nodes) que se encuentran en la carpeta /dev.

Para ver el tipo de archivo, utilice el comando ls.

Sintaxis

```
ls <opción> <archivos>
```

La opción utiliza:

<https://dogramcode.com/libros-sistemas>

Opción corta	Descripción
-l	Utiliza un formato de visualización largo.

El primer carácter de cada línea define el tipo de archivo:

```
# ls -l /etc

...

drwxr-xr-x 2 root root  4096 sept. 11 16:15 modules-load.d
-rw-r--r-- 1 root root   286 nov.  30  2014 motd
lrwxrwxrwx 1 root root    12 jun  15 19:00 mtab -> /proc/ mounts
...
*** Salida truncada ***
```

Sin duda ya conocemos los archivos de carpetas (d), normales (-) y enlaces (l).

Los dispositivos en modo bloque se identifican por una b:

```
#ls -l /dev/sd*

brw-rw---- 1 root disk 8,  0 oct.   4 12:20 /dev/sda
brw-rw---- 1 root disk 8,  1 oct.   4 12:20 /dev/sda1
brw-rw---- 1 root disk 8,  2 oct.   4 12:20 /dev/sda2
brw-rw---- 1 root disk 8,  5 oct.   4 12:20 /dev/sda5
brw-rw---- 1 root disk 8, 16 oct.   4 12:20 /dev/sdb
brw-rw---- 1 root disk 8, 17 oct.   4 12:20 /dev/sdb
```

Los dispositivos en modo carácter se identifican por una c:

```
#ls -l /dev/tty[1-6]

crw--w---- 1 root tty 4, 1 oct.   4 12:20 /dev/tty1
crw--w---- 1 root tty 4, 2 oct.   4 12:20 /dev/tty2
crw--w---- 1 root tty 4, 3 oct.   4 12:20 /dev/tty3
crw--w---- 1 root tty 4, 4 oct.   4 12:20 /dev/tty4
crw--w---- 1 root tty 4, 5 oct.   4 12:20 /dev/tty5
crw--w---- 1 root tty 4, 6 oct.   4 12:20 /dev/tty6
```

Estos archivos especiales se caracterizan por dos valores numéricos.

- El número mayor identifica el controlador.
- El número menor identifica el dispositivo. Un controlador puede gestionar múltiples dispositivos.

El archivo `/proc/devices` muestra los controladores de dispositivos configurados en el kernel en ejecución. Contiene dos secciones. La primera muestra los dispositivos en modo carácter:

```
#cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttyS
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 6 lp
 7 vcs
10 misc
13 input
21 sg
29 fb
99 ppdev
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
226 drm
252 hidraw
253 bsg
254 rtc
*** Salida truncada ***
```

La segunda muestra los dispositivos en modo bloque:

```
*** Salida truncada ***
```

```
Block devices:
```

```
259 blkext
```

```
7 loop
```

```
8 sd
```

```
11 sr
```

```
65 sd
```

```
66 sd
```

```
67 sd
```

```
68 sd
```

```
69 sd
```

```
70 sd
```

```
71 sd
```

```
128 sd
```

```
129 sd
```

```
130 sd
```

```
131 sd
```

```
132 sd
```

```
133 sd
```

```
134 sd
```

```
135 sd
```

```
254 device-mapper
```

La estructura del archivo se compone de dos columnas: el número mayor y el nombre del dispositivo.

El comando `lsblk` muestra los dispositivos de bloque.

Sintaxis

```
lsblk <opciones>
```

Opciones útiles:

<https://dogramcode.com/libros-sistemas>

Opción		Descripción
corta	larga	
-a	--all	Muestra todos los dispositivos en modo bloque.
-d	--nodeps	No muestra los elementos dependientes.
-f	--fs	Muestra información sobre el sistema de archivos.
-m	--perms	Muestra el usuario y el grupo propietario del dispositivo.
-l	--list	Visualización en formato lista.
-n	--noheadings	No muestra la línea de cabecera.
-r	--raw	Visualización en formato bruto.

Ver todos los dispositivos de bloque:

```
# lsblk -a
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	50G	0	disk	
sda1	8:1	0	50G	0	part	
SystemVG-rootLV (dm-0)	254:0	0	14G	0	lvm	/
SystemVG-swapLV (dm-1)	254:1	0	488M	0	lvm	[SWAP]
SystemVG-homeLV (dm-2)	254:2	0	35,4G	0	lvm	/home
SystemVG-optLV (dm-3)	254:3	0	168M	0	lvm	/opt
sr0	11:0	1	1024M	0	rom	
loop0	7:0	0		0	loop	
loop1	7:1	0		0	loop	
loop2	7:2	0		0	loop	
loop3	7:3	0		0	loop	
loop4	7:4	0		0	loop	
loop5	7:5	0		0	loop	
loop6	7:6	0		0	loop	
loop7	7:7	0		0	loop	

2. Controladores de red

Una interfaz de red puede ser un dispositivo de hardware o software como el bucle (loopback). Se encarga de enviar y recibir paquetes.

La interfaz no está mapeada en un nodo en la carpeta /dev. Por lo tanto, no hay ninguna entrada en el sistema de archivos. De modo que la comunicación entre el kernel y un controlador de una interfaz de red no funciona de la misma manera que con un dispositivo en modo bloque o en modo carácter. El kernel llama a las funciones relativas a la transmisión de paquetes.

Se asigna un nombre único, como ethx a la interfaz. El valor de x será 0 para la primera tarjeta, 1 para la segunda y así sucesivamente...

Las interfaces inalámbricas llevarán el nombre ethx o wlanx según la declaración definida en el sistema. El demonio udev lee los archivos del directorio /lib/udev/rules.d/ que contiene las reglas del sistema, y también los archivos del directorio /etc/udev/rules.d/ que pueden contener reglas escritas por el usuario root. Escribir las reglas permite, en caso necesario, modificar el nombre de una interfaz.

Todos los archivos se clasifican en función de su nombre por orden ascendente y luego se ejecutan.

CentOS tenía este mismo mecanismo de nombres hasta la versión 6. En la versión 7, el nombre de las interfaces de comunicación cambia. Tiene un prefijo de dos caracteres:

Prefijo	Tipo de interfaz
en	Ethernet
wl	Red local inalámbrica (WLAN)
ww	Red de área extensa inalámbrica (WWAN)

La continuación del nombre contiene el bus PCI y el slot.

Ejemplos:

La interfaz de cable en el bus 0 slot 1 tiene por nombre `enp0s1`.

La interfaz WLAN en el bus 1 slot 0 tiene por nombre `wlp1s0`.

Estudiaremos con más detalle la configuración de la red en el capítulo Mantenimiento de la configuración de red.

Librerías

Una librería (library) es un archivo que contiene uno o varios archivos objeto cuya extensión es `.o`. Estos últimos pueden contener funciones del lenguaje C o métodos y clases escritos en C++, que pueden ser invocadas por uno o varios programas. Se ejecutan en el anillo de protección 3.

Tenemos dos tipos de librerías:

- las librerías estáticas (static libraries).
- las librerías compartidas (dynamically linked shared object libraries).

1. Librerías estáticas

Una librería estática es un archivo con la extensión `.a`. Se copia en los programas que la utilizan durante la construcción de los mismos.

El comando `ar` gestiona los archivos y también permite mostrar el contenido de una librería. A diferencia de CentOS 7, no se instalará por defecto en las distribuciones Debian 8 y Ubuntu Server 14.04 LTS. Hay que instalar el paquete `binutils`.

```
# apt-get install binutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

```
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  binutils-doc
Se instalarán los siguientes paquetes NUEVOS:
  binutils
0 actualizados, 1 se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 2.076 kB de archivos.
Se utilizarán 11,7 MB de espacio de disco adicional después de esta
operación.
Des: 1 http://es.archive.ubuntu.com/ubuntu/ trusty/universe ksh amd64
93u+20120801-1 [1.583 kB]
Descargados 1.583 kB en 0seg. (2.147 kB/s)
Seleccionando el paquete binutils previamente no seleccionado.
Leyendo la base de datos ... 198862 ficheros o directorios instalados
actualmente.)
Preparing to unpack .../binutils_2.24-5ubuntu14_amd64.deb ...
Unpacking binutils (2.24-5ubuntu14) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Configurando binutils (2.24-5ubuntu14) ...
Processing triggers for libc-bin (2.19-0ubuntu6.7) ...
```

Sintaxis

```
ar <opción> <archivo>
```

Opción corta	Descripción
-t	Muestra el contenido de un archivo

Ejemplo con CentOS

Vamos a mostrar los archivos .o contenidos en la librería libc.a:

<https://dogramcode.com/libros-sistemas>

```
# ar -t /usr/lib/x86_64-redhat-linux6E/lib64/libc.a
init-first.o
libc-start.o
sysdep.o
version.o
check_fds.o
libc-tls.o
elf-init.o
dso_handle.o
errno.o
*** Salida truncada ***
```

2. Librerías compartidas

Una librería compartida es un archivo con la extensión .so. Debe estar presente durante la compilación de programas, especialmente durante la fase de la edición de los enlaces. No se incluye en el ejecutable sino que se vincula al mismo.

Existe una multitud de librerías estándar, como:

- GNU C standard (glibc)
- C++ standard (libstdc++)
- Math (libm)
- Tiempo Real (librt)
- Las librerías para el desarrollo de la interfaz de usuario en un entorno gráfico como Xlib, Motif, GTK+, Qt...

Si la librería compartida falta o está mal ubicada, el programa no funcionará.

3. Ubicación de las librerías

Según el FHS (Filesystem Hierarchy Standard), las librerías esenciales para los binarios situados en los archivos /bin y /sbin se encuentran en /lib. Las librerías para los binarios situados en los archivos /usr/bin y /usr/sbin se almacenan en /usr/lib.

Para Debian y Ubuntu, la carpeta /lib64 contiene los enlaces simbólicos a /lib/x86_64-linux-gnu:


```
# ls -l /lib64
lrwxrwxrwx 1 root root 32 mar 31 00:48 ld-linux-x86-64.so.2 ->
/lib/x86_64-linux-gnu/ld-2.19.so
```

He aquí el objeto del enlace:

```
# cd /lib/x86_64-linux-gnu/
drwxr-xr-x 2 root root 4096 jun 15 18:29 device-mapper
-rwxr-xr-x 1 root root 140928 ago 31 00:48 ld-2.19.so
lrwxrwxrwx 1 root root 10 ago 31 00:48 ld-linux-x86-64.so.2 ->
ld-2.19.so
lrwxrwxrwx 1 root root 15 sep 8 2014 libacl.so.1 ->
libacl.so.1.1.0
-rw-r--r-- 1 root root 35288 sep 8 2014 libacl.so.1.1.0
-rw-r--r-- 1 root root 14960 ago 31 00:48 libanl-2.19.so
*** Salida truncada ***
```

En cuanto a CentOS 7, las carpetas /lib y /lib64 mantienen enlaces simbólicos.

```
# ls -l /lib*
lrwxrwxrwx. 1 root root 7 18 dic 2014 /lib -> usr/lib
lrwxrwxrwx. 1 root root 9 18 dic 2014 /lib64 -> usr/lib64
```

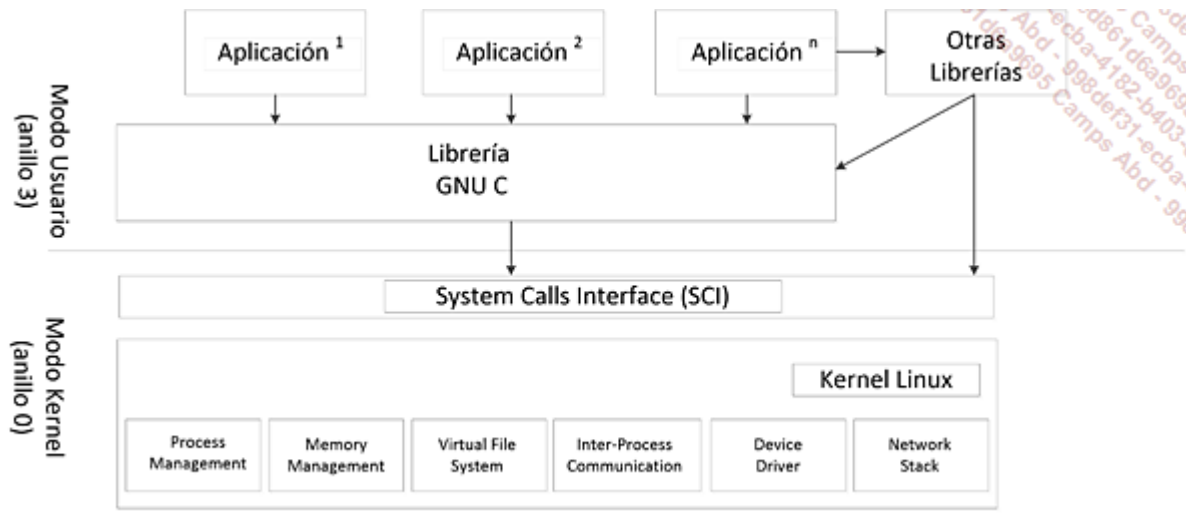
Al igual que las carpetas /bin y /sbin:

```
# ls -l /*bin
lrwxrwxrwx. 1 root root 7 18 dic 2014 /bin -> usr/bin
lrwxrwxrwx. 1 root root 8 18 dic 2014 /sbin -> usr/sbin
```

Lo interesante es que la carpeta /usr puede ser montada por initramfs.

Llamadas de sistema

Las aplicaciones se comunican con el kernel a través de las llamadas de sistema (system calls o syscalls).



Una aplicación llama a las funciones de una librería como glibc la cual, empleando a su vez a la interfaz de llamadas de sistema (SCI o System Calls Interface), pide al núcleo las tareas por cuenta de la aplicación.

El sistema GNU/Linux ofrece más de 200 llamadas de sistema agrupadas en categorías:

- Gestión de procesos
- Gestión de señales
- Gestión del sistema de archivos
- Mecanismo de protección
- Funciones de temporizador y estadísticas

Cuando una aplicación usa, por ejemplo, la función `printf()` para mostrar una cadena de caracteres con formato en una consola, invoca a la función `write()`.

La sección 2 de las páginas del manual proporciona una descripción de las llamadas de sistema.

```
# man 2 intro
```

He aquí el resultado en pantalla.

```
Aplicaciones Lugares Terminal
root@localhost:~/Escritorio

Archivo Editar Ver Buscar Terminal Ayuda
INTRO(2) Manual del programador de Linux INTRO(2)

NOMBRE
intro - Introducción a las llamadas al sistema

DESCRIPCIÓN
Este capítulo describe las llamadas al sistema de Linux. En
syscalls(2) encontrará una lista de las 164 llamadas al sistema pre-
sentes en Linux 2.0.

Llamadas directas
En la mayoría de los casos no es necesario invocar una llamada al sis-
tema directamente, pero hay veces en que la biblioteca C estándar
carece de una función adecuada.

Síntesis
#include <linux/unistd.h>

Una macro de tipo _syscall

Configuración
Lo importante a conocer sobre una llamada al sistema es su prototipo.
Es necesario saber cuántos argumentos usa, sus tipos, y el tipo retor-
nado por la función. Hay seis macros que facilitan la llamada real al
sistema. Tienen la siguiente forma:

_syscall(x,tipo,nombre,tip0,arg1,tipo2,arg2...)

donde x es 0-5, el número de argumentos usado por la lla-
mada del sistema

tipo es el tipo devuelto por la llamada al sistema

nombre es el nombre de la llamada al sistema

tip0 es el tipo del argumento nº 0

argN es el nombre del argumento nº N

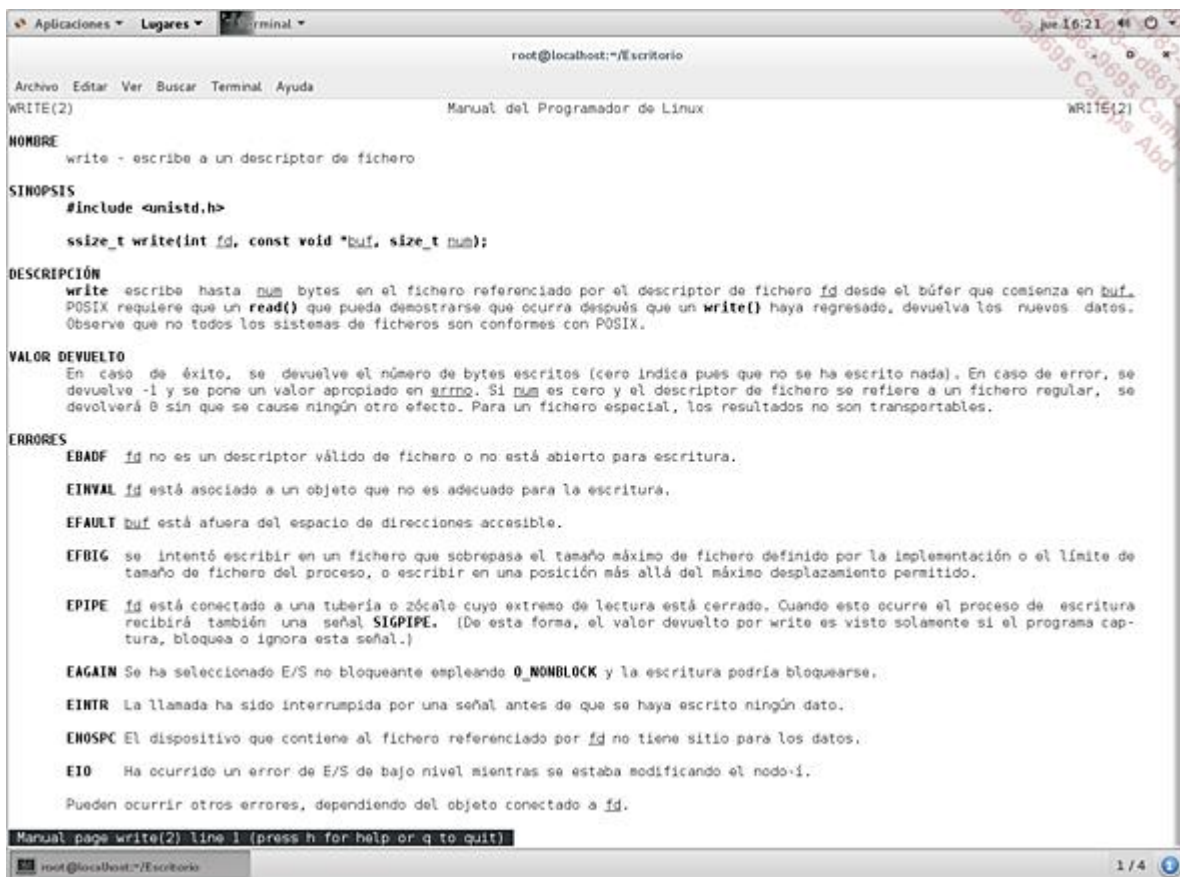
Estas macros crean una función llamada nombre con los argumentos
especificados. Una vez incluida _syscall() en el fichero fuente, se
invoca a la llamada al sistema mediante nombre.

EJEMPLO
Manual page intro(2) line 1 (press h for help or q to quit)

root@localhost:~/Escritorio 1 / 4
```

Si escribe man 2 write, accederá a la página del manual de la función write():

<https://dogramcode.com/libros-sistemas>



Shell

Un shell es un programa que sirve de interfaz entre el usuario y el sistema operativo. Existen dos formas distintas:

- El intérprete de líneas de comandos (CLI o Command Line Interface).
- El shell gráfico (GUI o Graphical User Interface) del que hablaremos más adelante en este capítulo.

1. Diferentes shells

Existen varios shells CLI para GNU/Linux. La elección de un shell en lugar de otro depende generalmente de las necesidades, es decir, las funcionalidades que ofrece, de su tamaño en disco duro o en la RAM.

a. Bourne Shell

El Bourne shell, diseñado por Stephen Bourne, era el shell por defecto de Unix versión 7 en 1977. Reemplazó al Thompson Shell de Kenneth Lane Thompson, el creador de Unix. El archivo ejecutable lleva el mismo nombre: sh.

Dado el hecho de que Bourne shell no es open source, el archivo /bin/sh es un enlace simbólico al Dash (Debian Almquist Shell) en las distribuciones Debian y Ubuntu.

Para comprobar a qué corresponde sh, teclee lo siguiente:

```
#ls -l $(which sh)
lrwxrwxrwx 1 root root 4 nov. 8 2014 /bin/sh -> dash
```

En cuanto a CentOS, el vínculo simbólico apunta a Bash (Bourne Again Shell):

```
#ls -l $(which sh)
lrwxrwxrwx. 1 root root 4 16 sept. 12:46 /usr/bin/sh -> bash
```

b. Almquist Shell y Debian Almquist Shell

Almquist Shell (A Shell o Ash) es un shell derivado del Bourne shell desarrollado por Kenneth Almquist. Siendo poco goloso, rápido y compatible con la norma POSIX, está integrado en BusyBox. Este último proporciona un conjunto de comandos estándar de Unix en un único ejecutable que se pueden utilizar, entre otros, en sistemas embebidos.

Debian Almquist Shell (DASH) sustituye a Ash en Debian 5 Lenny y luego se convierte en el shell por defecto de Debian Squeeze 6 y Ubuntu 6.10 Edgy EFT.

Dash estuvo presente en la distribución CentOS 6 pero ya no se instala por defecto en CentOS 7.

c. Korn Shell

Korn Shell (ksh), desarrollado por David Korn a comienzos de los años 1980, es compatible con el Bourne Shell y se convierte en el shell por defecto de muchos Unix como HP UX, IBM AIX, Sun Solaris... Se ajusta a la norma POSIX 1003.2.

No se instala por defecto. Debian, Ubuntu y CentOS ofrecen la instalación del Korn Shell 93 (ksh93).

Ejemplo con Debian o Ubuntu

La instalación se realiza con el comando apt-get install:

```
# apt-get install ksh
Leyendo lista de paquetes...
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  ksh

0 actualizados, 1 se instalarán, 0 para eliminar y 1 no actualizados.
Necesito descargar 1.583 kB de archivos.
Se utilizarán 3.229 kB de espacio de disco adicional después de esta
operación.
Des: 1 http://es.archive.ubuntu.com/ubuntu/ trusty/universe ksh amd64
93u+20120801-1 [1.583 kB]
Descargados 1.583 kB en 0seg. (2.147 kB/s)
Seleccionando el paquete ksh previamente no seleccionado.
Leyendo la base de datos ... 199134 ficheros o directorios instalados
actualmente.)
Preparing to unpack .../ksh_93u+20120801-1_amd64.deb ...
Unpacking ksh (93u+20120801-1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Configurando ksh (93u+20120801-1) ...
update-alternatives: utilizando /bin/ksh93 para proveer /bin/ksh (ksh)
en modo automático
```

Localice el programa Korn Shell 93 con which:

```
# which ksh ksh93
/usr/bin/ksh
/bin/ksh93
```

/usr/bin/ksh es un enlace simbólico:

```
# ls -l /usr/bin/ksh
lrwxrwxrwx 1 root root      29 abr 29 20:26 /usr/bin/ksh ->
/etc/alternatives/usr.bin.ksh
```

Apunta al archivo /etc/alternatives/ksh que apunta a su vez a /bin/ksh93:

```
# ls -l /etc/alternatives/ksh
lrwxrwxrwx 1 root root 10 jun  2 18:04 /etc/alternatives/ksh ->
/bin/ksh93
```

Para proceder a su ejecución, ejecute ksh o ksh93:

```
root@debian:~# ksh
#
```

Salga del Korn Shell tecleando exit:

```
# exit
root@debian:~#
```

Ejemplo con CentOS

La instalación se realiza con el comando yum install:

```
# yum -y install ksh
Complementos cargados:fastestmirror, langpacks
base | 3.6 kB | 00:00
extras | 3.4 kB | 00:00
updates | 3.4 kB | 00:00
(1/4): extras/7/x86_64/primary_db | 117 kB | 00:00
(2/4): updates/7/x86_64/primary_db | 4.1 MB | 00:01
(3/4): base/7/x86_64/group_gz | 155 kB | 00:01
(4/4): base/7/x86_64/primary_db | 5.3 MB | 00:18
Determining fastest mirrors
* base: ftp.pasteur.fr
* extras: mirror0.babylon.network
* updates: fr2.rpmfind.net
```

Resolviendo dependencias

--> Ejecutando prueba de transacción

---> Paquete ksh.x86_64 0:20120801-22.el7_1.3 debe ser instalado

--> Resolución de dependencias finalizada

Dependencias resueltas

```
=====
Package      Arquitectura  Versión      Repositorio  Tamaño
=====
Instalando:
ksh          x86_64       20120801-22.el7_1.3  updates      881 k
```

Resumen de la transacción

```
=====
Instalar 1 Paquete
Tamaño total de la descarga: 881 k
Tamaño instalado: 3.1 M
Downloading packages:
ksh-20120801-22.el7_1.2.x86_64.rpm | 880 kB      00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalación : ksh-20120801-22.el7_1.2.x86_64      1/1
  Verificación : ksh-20120801-22.el7_1.2.x86_64      1/1
```

Instalado:

ksh.x86_64 0:20120801-22.el7_1.3

¡Listo!

Localice el programa Korn Shell 93 con which:

```
# 11 $(which ksh ksh93)
```



```
lrwxrwxrwx. 1 root root      21 29 sept. 20:29 /usr/bin/ksh ->
/etc/alternatives/ksh
-rwxr-xr-x. 1 root root 1514808  6 ago   01:55 /usr/bin/ksh93
```

Podemos constatar aquí arriba que `/usr/bin/ksh` es un enlace simbólico. Apunta al archivo `/etc/alternatives/ksh` que apunta a su vez a `/usr/bin/ksh93`.

Para proceder a su ejecución, teclee `ksh` o `ksh93`.

d. C-Shell y Tenex C-Shell

El C-shell (`csh`), diseñado por William N. Joy (alias Bill Joy), es una evolución del shell `sh` utilizando una sintaxis más cercana al lenguaje C utilizado en primer lugar por Unix BSD (Berkeley Software Distribution).

Tenex CSH (`tcsh`), basado en C-Shell, aporta funcionalidades adicionales como la edición directa de la línea de comandos.

Ejemplo con Debian y Ubuntu

Debian y Ubuntu disponen de dos paquetes: `csh` y `tcsh`.

Instalación de `csh`:

```
# apt-get install csh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  csh

0 actualizados, 1 se instalarán, 0 para eliminar y 1 no actualizados.
Necesito descargar 236 kB de archivos.
*** Salida truncada ***
```

Instalación de `tcsh`:

```
# apt-get install tcsh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

```
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
    tcsh
```

```
0 actualizados, 1 se instalarán, 0 para eliminar y 1 no actualizados.
Necesito descargar 458 kB de archivos.
*** Salida truncada ***
```

Localice los dos programas:

```
# ll $(which csh tcsh)
lrwxrwxrwx 1 root root      21 sept. 29 20:55 /bin/csh ->
/etc/alternatives/csh
-rwxr-xr-x 1 root root 395504 sept.  9 2014 /usr/bin/tcsh
```

/bin/csh es un enlace simbólico que apunta al archivo /etc/alternatives/csh siendo él mismo un enlace simbólico que apunta a /bin/bsd-csh:

```
# ls -l /etc/alternatives/csh
lrwxrwxrwx 1 root root 12 sept. 29 20:57 /etc/alternatives/csh ->
/bin/bsd-csh
```

El comando `which` nos informa de que `tcsh` se encuentra en `/usr/bin`. Sin embargo, podemos apreciar que existe un enlace `tcsh` en la carpeta `/bin` que apunta a `/usr/bin/tcsh`:

```
# ls -l /bin/tcsh
lrwxrwxrwx 1 root root 13 jun 21 2012 /bin/tcsh -> /usr/bin/tcsh
```

Ejemplo con CentOS

CentOS, por su parte, ofrece un paquete `csh`, que en realidad es `tcsh`. Se instala por defecto. Con `which`, localizamos `csh` y `tcsh`:

```
# ll $(which csh tcsh)
lrwxrwxrwx. 1 root root      4  4 oct.  13:45 /usr/bin/csh -> tcsh
-rwxr-xr-x. 1 root root 404416 10 jun   2014 /usr/bin/tcsh
```

Comprobamos más arriba que efectivamente csh es un enlace simbólico que apunta a tcsh.

e. Bourne Again Shell

Linux propone por defecto el shell Bash creado por Brian Fox en 1988 y mantenido en la actualidad por Chet Ramey. Es compatible con el Bourne Shell e incorpora también funcionalidades procedentes del Korn Shell y C-Shell.

Como todos los proyectos de desarrollo, Bash evoluciona para corregir sus errores, sus vulnerabilidades, pero también para aportar nuevas funcionalidades. Esto puede introducir algunas incompatibilidades con las versiones anteriores.

Para ver la versión de shell, vamos a usar el comando echo y la variable de entorno \$BASH_VERSION:

```
# echo $BASH_VERSION
4.3.30(1)-release
```

En el cuadro siguiente, tenemos la versión de Bash utilizada en cada distribución.

Distribución	Versión
Debian 8	4.3.30(1)-release
Ubuntu Server 14.04 LTS	4.3.11(1)-release
CentOS 7	4.2.46(1)-release

2. ¿Cuál es el shell de login?

El shell asociado a una cuenta de usuario se encuentra en /etc/passwd. El comando grep permite buscar el shell para un usuario:

Sintaxis

```
grep <patrón_de_búsqueda> <archivo>
```

Ejemplo

```
# grep bob /etc/passwd  
bob:x:1000:1000:bob,,,:/home/bob:/bin/bash
```

Bob utiliza bash como shell de login.

Los shells de conexión válidos dentro del sistema operativo se encuentran en el archivo `/etc/shells`. Por supuesto, su contenido varía en función de los shells instalados.

Debian

```
# cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
bin/dash  
/bin/bash  
/bin/rbash  
/bin/ksh93  
/bin/tcsh  
/usr/bin/tcsh  
/bin/csh
```

Ubuntu

```
# cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
/bin/dash
```

```
/bin/bash
/bin/rbash
/usr/bin/tmux
/usr/bin/screen
/bin/ksh93
```

CentOS

```
# cat /etc/shells
/bin/sh
bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/bin/tcsh
/bin/csh
/bin/ksh
/bin/rksh
```

3. Páginas del manual

a. Secciones

Las páginas de manual, conocidas bajo el nombre de «man pages», constituyen la documentación de referencia para el usuario, el administrador y el programador. Contienen todos los comandos y funciones disponibles y se organizan en nueve secciones:

Sección	Descripción
1	Comandos de usuario

2	Llamadas del sistema
3	Funciones de librerías (con la excepción de las funciones de llamadas del sistema)
4	Archivos especiales
5	Formato de archivos
6	Juegos
7	Varios
8	Comandos de administración y mantenimiento
9	Llamadas al kernel de Linux

La sección 9 no se instala de manera estándar en las distribuciones. Hay que añadir en Debian, el paquete linux-manual. Lo encontraremos tecleando el comando `apt-cache search ^linux-manual`. Luego, se instala empleando el comando `apt-get install <nombre del paquete>`.

El comando `man` permite consultar las páginas del manual.

Sintaxis

```
man <número_de_sección> <nombre_del_comando>
```

Cada sección presenta una página de inicio cuya palabra clave es `intro`:

```
# man 1 intro
```

El resultado nos devuelve por pantalla la página principal de la sección 1:

```
javier@debian8: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
INTRO(1)                                Linux User's Manual                                INTRO(1)  
  
NAME  
    intro - introduction to user commands  
  
DESCRIPTION  
    Section 1 of the manual describes user commands and tools, for example,  
    file manipulation tools, shells, compilers, web browsers, file and  
    image viewers and editors, and so on.  
  
    All commands yield a status value on termination. This value can be  
    tested (e.g., in most shells the variable $? contains the status of  
    the last executed command) to see whether the command completed suc-  
    cessfully. A zero exit status is conventionally used to indicate suc-  
    cess, and a nonzero status means that the command was unsuccessful.  
    (Details of the exit status can be found in wait(2).) A nonzero exit  
    status can be in the range 1 to 255, and some commands use different  
    nonzero status values to indicate the reason why the command failed.  
  
NOTES  
    Linux is a flavor of UNIX, and as a first approximation all user com-  
    mands under UNIX work precisely the same under Linux (and FreeBSD and  
    lots of other UNIX-like systems).  
Manual page intro(1) line 1 (press h for help or q to quit)
```

En las esquinas superior izquierda y derecha de la pantalla veremos INTRO(1) que nos situará en las páginas del manual: página intro de la sección 1.

b. Buscar la ayuda de un comando

La búsqueda de una descripción o del número de la sección de un comando se realiza con la opción -k del comando man o bien el comando apropos.

Sintaxis

```
man -k <palabra_clave>
```

o:

```
apropos <palabra_clave>
```

La palabra clave se trata como una expresión regular. Puede ser necesario el uso de caracteres especiales:

- `^` significa «comenzando por».
- `$` significa «termina con».

Ejemplo

Queremos ver las páginas del manual para el comando passwd. Vamos a omitir los caracteres especiales.

```
# man -k passwd
chpasswd (8)          - update passwords in batch mode
gpasswd (1)           - Administer /etc/group and /etc/gshadow
passwd (1)            - update user's authentication tokens
passwd (5)            - the password file
update-passwd (8)     - Actualiza /etc/passwd, /etc/shadow y
                        /etc/group de for...
vipw (8)              - edit the password, group, shadow-password
                        or shadow-gr...
chgrpPASSWD (8)       - update group passwords in batch mode
Crypt::PasswdMD5 (3pm) - Provides interoperable MD5-based crypt()
                        functions
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)        - get passwd file entry reentrantly
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
lppasswd (1)          - add, change, or delete digest passwords.
mksmbpasswd (8)       - formats a /etc/passwd entry for a smbpasswd
                        file
pam_localuser (8)     - require users to be listed in /etc/passwd
passwd (1ssl)         - compute password hashes
passwd2des (3)        - RFS password encryption
smbpasswd (5)         - The Samba encrypted password file
smbpasswd (8)         - change a user's SMB password
```

Comprobamos que tenemos todas las líneas que contienen la palabra clave passwd. Ahora, ejecutamos el mismo comando añadiendo los caracteres especiales:

```
# man -k ^passwd$
passwd (1)            - update user's authentication tokens
```


passwd (5)	- the password file
passwd (1ssl)	- compute password hashes

Hemos obtenido la descripción:

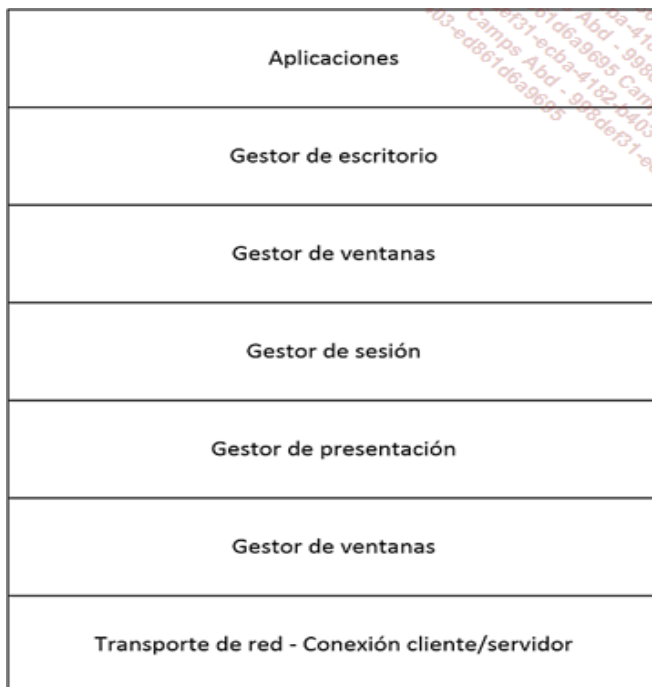
- del comando passwd que gestiona la contraseña de un usuario en la sección 1 (comandos de usuario).
- de la estructura del archivo /etc/passwd en la sección 5 (formatos de archivo).
- del cálculo de una contraseña. La sección 1ssl está asociada con OpenSSL.

Interfaz gráfica

La interfaz gráfica se instalará por defecto o no dependiendo de la distribución de GNU/Linux.

1. Capas gráficas

La arquitectura de la interfaz gráfica es un conjunto de aplicaciones que se comunican siguiendo un modelo cliente-servidor. Podemos esquematizar el funcionamiento en siete capas:



a. Transporte de red

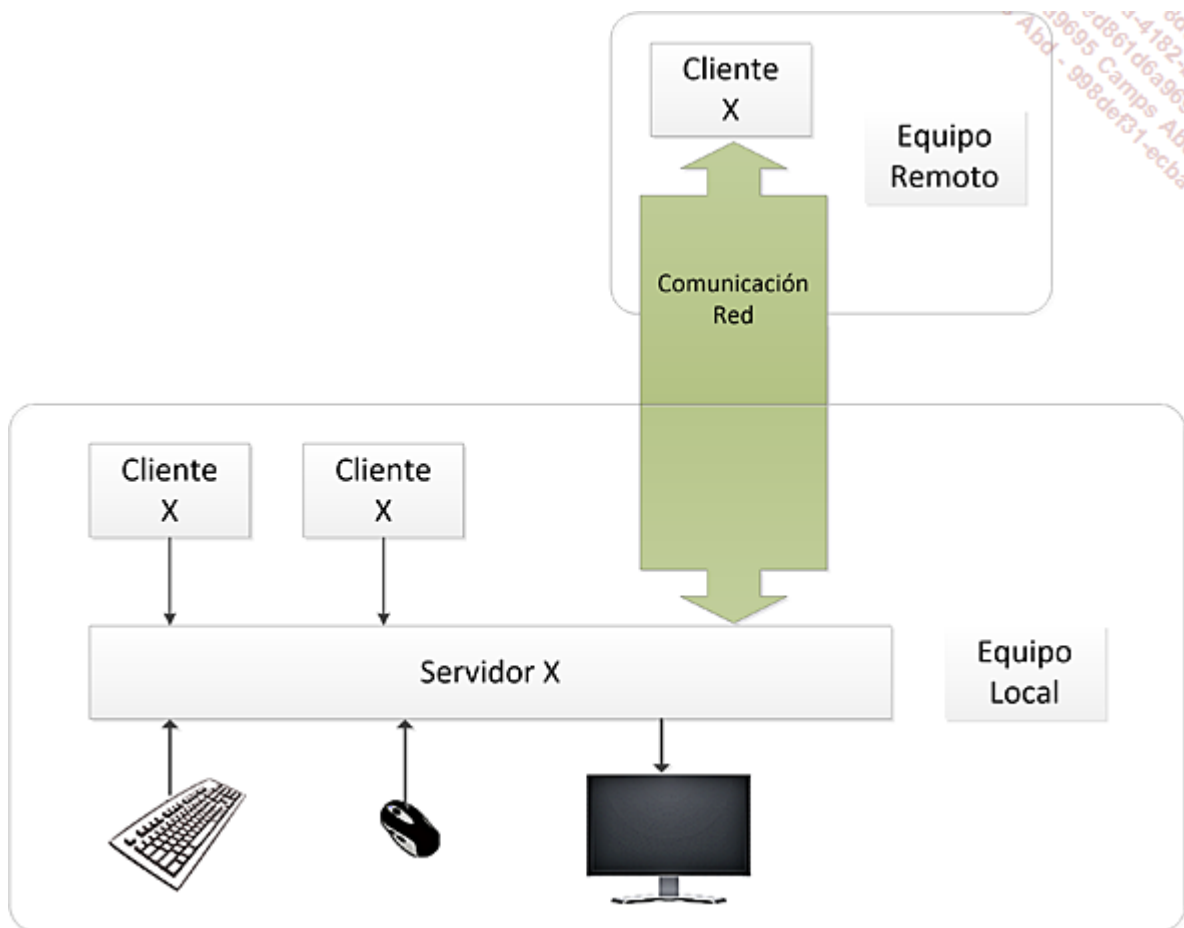
Esta capa permite la comunicación cliente-servidor a través de un protocolo como TCP/IP versión 4 o 6. Pueden usarse otros protocolos como IPX/SPX.

b. Sistema de ventanas

El sistema de ventanas implementado en los sistemas operativos tipo UNIX (Unix Like) es X Window System (conocido también bajo los nombres X11, X, X Versión 11). Se produjo una reacción muy controvertida por la comunidad de software libre debido a su licencia. Las distribuciones GNU/Linux adoptaron XFree86 que es una implementación libre de X Window System.

En 2004, la versión 4.4.0 de XFree86 cambió de licencia volviéndose por lo tanto incompatible con la licencia GPL (GNU General Public License). La mayoría de las distribuciones GNU/Linux se han orientado desde entonces hacia X.Org, una rama separada de XFree86.

X Window System se compone de un servidor X y uno o varios clientes X.



El servidor X gestiona las peticiones de visualización, el desplazamiento del ratón y la introducción de texto a través del teclado.

Un cliente X se sitúa en el mismo equipo que el servidor X o en un equipo remoto. Se conecta al servidor X para transmitir las peticiones empleando el protocolo X.

Ya hemos hablado de librerías en este capítulo. La programación X se basa en librerías:

La librería Xlib contiene funciones de bajo nivel escritas en lenguaje C para la parte cliente del protocolo X con el fin de comunicarse con el servidor X. Pocos programas recurren a ella directamente. Utilizan librerías como Xaw, Motif, Gtk+ o Qt, las cuales se basan en Xlib.

Xt (X toolkit), escrita también en lenguaje C, se basa en la librería Xlib y ofrece una interfaz de programación (API - Application Programming Interface) más fácil para los programadores.

Xaw (X Athena Widgets) ofrece un conjunto de componentes de interfaz gráfica (widgets) que permiten la programación de la interfaz del usuario.

Motif (IEEE 1295) permite a su vez, al igual que Xaw, desarrollar una interfaz de usuario en un entorno X.

Se tiende a pensar que hoy en día estos últimos resultan obsoletos. GTK+ (The GIMP Toolkit) y Qt (pronunciado «cute», no la letra Q y luego la letra T) los van reemplazando de forma progresiva.

c. Gestor de pantalla

El gestor de pantalla (display manager) estándar de X Window System es XDM (X Display Manager). Este programa permite iniciar una sesión en un servidor X local o remoto. Otros gestores de pantalla están desarrollados y ofrecen a veces funcionalidades adicionales.

La siguiente tabla enumera los principales gestores de pantalla:

Gestor de pantalla	Descripción
XDM	Gestor por defecto de X Window System.
GDM	Gestor por defecto del entorno gráfico GNOME.
KDM	Gestor por defecto del entorno gráfico KDE.
LightDM	Gestor de pantalla por defecto de la distribución Ubuntu Desktop 11.10 (Oneiric) y posteriores para sustituir a GDM.
LDM	Gestor de pantalla para Linux Terminal Server Project (LTSP).
LXDM	Entorno gráfico Lightweight X11 Desktop Environment (LXDE)
MDM	Gestor de pantalla de Linux Mint basado en GDM 2.20.

d. Gestor de sesión

El gestor de sesión (session manager) permite a los usuarios abrir y cerrar una sesión guardando su estado para la próxima conexión. Las aplicaciones y los scripts del usuario pueden ejecutarse durante el inicio de sesión.

Varía en función del entorno de escritorio o de la filosofía de la distribución.

e. Gestor de ventanas

El gestor de ventanas (window manager) es sin duda la parte más importante del entorno de escritorio. Define el aspecto y el comportamiento de las ventanas y, en particular, la barra de título, los bordes, el tamaño, el desplazamiento, los colores. Proporciona también las barras de tareas, un anclaje para las aplicaciones, los menús, los escritorios virtuales, el fondo para el escritorio, etc.

Los entornos de escritorio incorporan por defecto un gestor de ventanas:

Gestor de ventanas	Entorno de escritorio
metacity	GNOME 2
mutter	GNOME3
compiz-fusion	UNITY
mate-windows-manager	MATE
kwin	KDE
muffin	CINNAMON
xfwm4	XFCE
openbox	LXDE

Por supuesto, existen otros gestores.

En la mayoría de los casos podemos cambiar de un gestor de ventanas a otro. Por ejemplo, podemos reemplazar metacity por compiz-fusion para que GNOME 2 se beneficie de un escritorio con efectos gráficos 3D utilizando la aceleración por hardware de la tarjeta gráfica.

El GNOME Shell de GNOME 3 gestiona el panel de control, el área de notificación y el selector de ventanas. Está estrechamente ligado con el gestor de ventanas muttery, de modo que es difícil de cambiar. En este caso, debemos efectuar este procedimiento:

Instalamos compiz-fusion y nos desconectamos.

Volvemos a conectar seleccionando una sesión «Classic GNOME with Compiz».

Lo mismo puede decirse de Ubuntu Desktop, el entorno de escritorio Unity fue desarrollado como plug-in de compiz.

f. Gestor de escritorio

El rol de un gestor de escritorio es implementar una interfaz gráfica para que el usuario pueda comunicarse con el equipo. La mayoría de estos entornos adoptan el paradigma de escritorio. El usuario trabaja en un escritorio virtual donde podemos colocar los iconos.

El paradigma de escritorio se va abandonando cada vez más hacia una nueva interfaz que privilegia el uso de espacios de trabajo múltiples y hace hincapié en la simple transición de una tarea a otra.

Hemos hablado de algunos nombres de gestores de escritorio (desktop environment) en este capítulo. En cualquier momento podemos añadir o quitar un entorno, e incluso sustituirlo por otro.

En el siguiente cuadro tenemos el entorno de escritorio básico para las distribuciones GNU/Linux a las que nos referimos en este libro:

Distribución GNU/Linux	Descripción
Debian 8	GNOME 3
Ubuntu Server 14.04 LTS	Sin entorno gráfico
CentOS 7	GNOME 3



En Debian y CentOS podemos elegir si la interfaz gráfica debe ser instalada o no durante su instalación.

2. Desinstalar un entorno gráfico

Una interfaz gráfica se destina por lo general a estaciones de trabajo en lugar de a los servidores. En estos últimos podemos considerar que es una capa adicional de aplicación que consume algo de memoria y que ocupa espacio en disco. Este no es el punto más importante. Desde el punto de vista de la seguridad, tenemos procesos adicionales que pueden añadir algunas vulnerabilidades. Desinstalar una interfaz reduce el área de exposición.

Vamos a ver cómo desinstalarla de un servidor Debian 6 o 7 y CentOS 5.

Debian

```
# apt-get autoremove --purge gnome*
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota: selección de libgnome-keyring-dev para la expresión
racional "gnome*"
...
```

Y luego:

```
# apt-get autoremove
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  accountsservice aisleriot apache2.2-bin apg argyll at-spi2-core
  baobab
  bogofilter bogofilter-bdb bogofilter-common brasero-common
*** Salida truncada ***
```

CentOS

La primera etapa consiste en eliminar el entorno de escritorio GNOME.

```
# yum groupremove "GNOME Desktop Environment" -y
Loaded plugins: fastestmirror, security
Setting up Group Process
Loading mirror speeds from cached hostfile
```

```
* base: mirror.ovh.net
* extras: mirror.ovh.net
* updates: mirrors.ircam.fr
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager-gnome.i386 1:0.7.0-13.el5 set to be
erased
---> Package alacarte.noarch 0:0.10.0-1.fc6 set to be erased
*** Salida truncada ***
```

En la segunda etapa, eliminamos X Window System.

```
# yum groupremove "X Window System" -y
Loaded plugins: fastestmirror, security
Setting up Group Process
Loading mirror speeds from cached hostfile
* base: mirror.ovh.net
* extras: mirror.ovh.net
* updates: mirrors.ircam.fr
Resolving Dependencies
--> Running transaction check
---> Package authconfig-gtk.i386 0:5.3.21-7.el5 set to be erased
---> Package bitmap-fonts.noarch 0:0.3-5.1.1 set to be erased
--> Processing Dependency: bitmap-fonts for package: vte
*** Salida truncada ***
```

Servicios y niveles de ejecución

Un servicio (daemon o demonio) es un programa que se inicia cuando se carga el sistema operativo y que se ejecuta en segundo plano. Estos programas pueden ser servidores web, de correo, de base de datos, un cortafuegos, entre otras cosas...

Antes, los servicios se arrancaban empleando sysVinit y su archivo de configuración /etc/inittab y se disponía de varios niveles de ejecución. Las versiones 1 a 5 de Debian y CentOS utilizaban ese mecanismo.

Scott James Remnant (Canonical Ltd) desarrolló upstart para sustituir el daemon sysinit (System V initialization). Se encuentra integrado por defecto a partir de la distribución Ubuntu 9.10 (Karmic Koala). Debian 6 y 7, así como CentOS 6, han integrado a su vez upstart.

Debian 8 y CentOS 7 utilizan ahora systemd para gestionar los servicios. En cuanto a Ubuntu Server 14.04 LTS, sigue utilizando upstart. systemd estaría previsto a partir de la versión 15.04. Podemos considerar que la próxima versión LTS, en este caso la 16.04, será por defecto systemd.

Resumiendo:

Distribución	Gestión de servicios
Debian 8	systemd
Ubuntu 14.04 LTS	upstart
CentOS 7	systemd

1. Gestión de servicios con Debian

Debian 7 incorpora ahora systemd. El archivo `/etc/inittab` falta, y el nivel de ejecución por defecto es 5 y no 2 como antes.

a. Mostrar el estado de un servicio

Podemos invocar el nombre del script que inicia el servicio con el argumento `status` o utilizar el comando `service`.

Sintaxis

```
servicio <nombre_del_servicio> <opción>
```

Opción necesaria:

Opción	Descripción
status	Muestra el estado del servicio especificado

Ejemplo

Deseamos conocer el estado del servicio `ssh`:

```
# service ssh status
ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
```



```
Active: active (running) since dim. 2015-10-04 12:20:20 CEST;
4h 38min ago
Main PID: 592 (sshd)
CGroup: /system.slice/ssh.service
└─592 /usr/sbin/sshd -D

oct. 04 12:20:20 Debsrv sshd[592]: Server listening on 0.0.0.0 port 22.
oct. 04 12:20:20 Debsrv sshd[592]: Server listening on :: port 22.
oct. 04 12:26:37 Debsrv sshd[1130]: Accepted password for bob from
192.168.56.1 port 46021 ssh2
oct. 04 12:26:37 Debsrv sshd[1130]: pam_unix(sshd:session): session
opened for user bob by (uid=0)
```

Observamos que el servicio ssh está arrancado, en caso contrario hubiéramos obtenido este mensaje:

```
# /etc/init.d/ssh status
ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
Active: inactive (dead) since dim. 2015-10-04 17:03:17 CEST; 7s ago
Process: 1903 ExecStart=/usr/sbin/sshd -D $SSH_OPTS (code=exited,
status=0/SUCCESS)
Main PID: 1903 (code=exited, status=0/SUCCESS)

oct. 04 17:01:47 Debsrv sshd[1903]: Server listening on 0.0.0.0 port 22.
oct. 04 17:01:47 Debsrv sshd[1903]: Server listening on :: port 22.
```

b. Ver el estado de todos los servicios

El comando `insserv` permite mostrar todos los servicios.

Sintaxis

```
insserv <opción>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-s	--showall	Ver el estado de todos los servicios.

Ejemplo

```
# insserv -s
K:04:0 1 6:rsyslog
:08:0 6:umountfs
K:05:0 6:umountnfs.sh
K:03:0 6:sendsigs
K:07:0 6:hwclock.sh
*** Salida truncada ***
S:02:2 3 4 5:ssh
*** Salida truncada ***
S:05:S:lvm2
S:11:S:procps
S:17:S:kbd
S:19:S:x11-common
S:08:S:kmod
S:11:S:udev-finish
S:01:2 3 4 5:sudo
```

Si tomamos la siguiente línea:

```
S:02:2 3 4 5:ssh
```

Sabemos que el servicio ssh arranca (S) con una prioridad 02 en los niveles de ejecución 2, 3, 4, 5.

c. Arrancar/detener un servicio

Podemos invocar el nombre del script para iniciar o detener el servicio o bien usar el comando `service`.

Sintaxis

```
service <nombre_del_servicio> <opción>
```

Opciones necesarias:

Opción	Descripción
start	Arranca el servicio.
stop	Detiene el servicio.
restart	Reinicia el servicio.

Ejemplos:

Para detener el servicio ssh:

```
# service ssh stop
```

Para arrancar el servicio ssh:

```
# service ssh start
```

Para reiniciar el servicio ssh:

```
# service ssh restart
```

d. Recargar la configuración de un servicio

La recarga de la configuración evita el reinicio del servicio.

Esta funcionalidad no está disponible para todos los servicios. Basta con ejecutar el script sin argumentos para comprobarlo. Si `reload` se encuentra en la lista de argumentos propuestos, la recarga puede efectuarse con este servicio.

Ejemplo

Para comprobar si el servicio ssh permite la recarga de su configuración.

```
# service ssh
[info] Usage: /etc/init.d/ssh {start|stop|reload|force-reload|
restart|try-restart|status}.
```

La opción reload se encuentra disponible:

```
# service ssh reload
```

e. Activar/desactivar los servicios del arranque

El comando que permite iniciar o no un servicio en Debian 8 es systemctl:

Sintaxis

```
systemctl <opción> <nombre_del servicio>
```

Opciones usadas:

Opción	Descripción
start	Arranca el servicio
stop	Detiene el servicio

Ejemplo: desactivar el servicio ssh durante el inicio

```
# systemctl stop ssh
```

Verifique:

```
# systemctl |grep ssh
```

El resultado muestra que el servicio volverá a arrancar en el próximo inicio (boot) del sistema.

Ejemplo: activar el servicio ssh durante el inicio

```
# systemctl start ssh
```

Verifique:

```
# systemctl |grep ssh
```

```
ssh.service          loaded active running   OpenBSD Secure Shell server
```

Ningún resultado, por lo tanto, el servicio no arrancará hasta el próximo inicio (boot) del sistema.

2. Gestión de servicios con Ubuntu Server

Scott James Remnant (Canonical Ltd) desarrolló upstart para sustituir el daemon sysinit. Se encuentra integrado por defecto a partir de la distribución Ubuntu 9.10 (Karmic Koala).

Su función es gestionar la ejecución de las tareas y los demonios al arrancar el sistema y también su interrupción cuando el equipo se apaga. Los supervisa mientras el sistema se encuentra funcionando.

Si los servicios están destinados a sysinit, se almacenan en /etc/rc2.d:

```
# ls /etc/rc2.d
README  S20rsync  S20screen-cleanup  S70dns-clean  S70pppd-dns
S99grub-common  S99ondemand  S99rc.local
```

En este caso, hay que utilizar los siguientes comandos:

- service
- update-rc.d

Si los servicios se convierten por upstart, debemos usar initctl. Se asocia también una estructura de archivos.

La carpeta /etc/init contiene los scripts de inicio para los servicios gestionados por upstart:

```
# ls -1
acpid.conf
apport.conf
atd.conf
*** Salida truncada ***
ssh.conf
*** Salida truncada ***
upstart-udev-bridge.conf
ureadahead.conf
ureadahead-other.conf
wait-for-state.conf
```

Por ejemplo, el script ssh.conf permite iniciar el servicio ssh.

```

# cat ssh.conf
# 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

# 'sshd -D' leaks stderr and confuses things in conjunction with
'console log'
console none

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

```

Ubuntu no utiliza el archivo `/etc/inittab` durante el arranque. No obstante, existe un archivo `/etc/init/rc.sysinit.conf` para garantizar la compatibilidad con `sysinit`:

```

# cat rc-sysinit.conf
# rc-sysinit - System V initialisation compatibility
#
# This task runs the old System V-style system initialisation scripts,
# and enters the default runlevel when finished.

Description    "System V initialisation compatibility"
author         "Scott James Remnant <scott@netsplit.com>"

start on (filesystem and static-network-up) or failsafe-boot
stop on runlevel

# Default runlevel, this may be overridden on the kernel command-line
# or by faking an old /etc/inittab entry
env DEFAULT_RUNLEVEL=2

emits runlevel

# There can be no previous runlevel here, but there might be old
# information in /var/run/utmp that we pick up, and we don't want
# that.
#
# These override that
env RUNLEVEL=
env PREVLEVEL=

console output
env INIT_VERBOSE

task

script

```

```

# Check for default runlevel in /etc/inittab
if [ -r /etc/inittab ]
then
    eval "$(sed -nre 's/^[^#][^:]*:([0-6sS]):initdefault:.*'/
DEFAULT_RUNLEVEL="\1";/p' /etc/inittab || true)"
fi
# Check kernel command-line for typical arguments
for ARG in $(cat /proc/cmdline)
do
    case "${ARG}" in
        -b|emergency)
            # Emergency shell
            [ -n "${FROM_SINGLE_USER_MODE}" ] || sulogin
            ;;
        [0123456sS])
            # Override runlevel
            DEFAULT_RUNLEVEL="${ARG}"
            ;;
        -s|single)
            # Single user mode
            [ -n "${FROM_SINGLE_USER_MODE}" ] || DEFAULT_RUNLEVEL=S
            ;;
    esac
done
# Run the system initialisation scripts
[ -n "${FROM_SINGLE_USER_MODE}" ] || /etc/init.d/rcS

# Switch into the default runlevel
telinit "${DEFAULT_RUNLEVEL}"
end script

```

Podemos apreciar que el nivel de ejecución se encuentra definido por la variable `env_Default runlevel=2`.

a. Mostrar el estado de un servicio

Podemos utilizar el comando service:

```
# service ssh status
ssh start/running, process 555
```

Sin embargo, tenemos el comando initctl que gestiona las tareas del sistema. En la carpeta /sbin se encuentra status, que es un enlace simbólico a initctl:

```
# ls -l /sbin/status
lrwxrwxrwx 1 root root 7 Ene 18 17:08 /sbin/status -> initctl
lrwxrwxrwx 1 root root 7 jul 18 2014 /sbin/status -> initctl
```

Sintaxis

```
initctl <acción> <nombre_del_servicio>
```

O bien:

```
status <nombre_del_servicio>
```

Las acciones.

Acción	Descripción	Enlace simbólico
status	Muestra el estado del servicio	/sbin/status

Ejemplo

```
# initctl status ssh
ssh start/running, process 555
```

O bien:

```
# status ssh
ssh start/running, process 555
```

b. Ver el estado de todos los servicios

Para conocer la lista de servicios y sus respectivos estados empleamos initctl o insserv.

Sintaxis

```
initctl <acción>
```

La acción:

Acción	Descripción
list	Muestra las tareas y sus estados.

Ejemplo

```
# initctl list
mountall-net stop/waiting
ountnfs-bootclean.sh start/running
passwd stop/waiting
rc stop/waiting
rsyslog start/running, process 495
*** Salida truncada ***
ssh start/running, process 846
*** Salida truncada ***
network-interface-container stop/waiting
ureadahead stop/waiting
```

c. Arrancar/detener un servicio

El comando `service` funciona de la misma manera que con la distribución Debian. Ubuntu propone `initctl`. Los enlaces simbólicos simplifican la sintaxis.

Sintaxis

```
initctl <acción> <nombre_del_servicio>
```

O bien:

```
<Enlace_simbólico> <nombre_del_sevice>
```

Opciones necesarias:

Acción	Descripción	Enlace simbólico
start	Arranca el servicio.	/sbin/start
stop	Detiene el servicio.	/sbin/stop
restart	Reinicia el servicio.	/sbin/restart

Ejemplos:

Para detener el servicio ssh:

```
# initctl stop ssh
ssh stop/waiting
```

O bien:

```
# stop ssh
ssh stop/waiting
```

Para arrancar el servicio ssh:

```
# initctl start ssh
ssh start/running, process 2098
```

O bien:

```
# start ssh
ssh start/running, process 2098
```

Para reiniciar el servicio ssh:

```
# initctl restart ssh
ssh start/running, process 2128
```

O bien:

```
# restart ssh
ssh start/running, process 2128
```

d. Recargar la configuración de un servicio

La recarga de la configuración se realiza con `initctl`.

Sintaxis

```
initctl <acción> <nombre_del_servicio>
```

O bien:

```
reload <nombre_del_servicio>
```

Opción necesaria:

Acción	Descripción	Enlace simbólico	Ejemplo
reload	Recargar la configuración del servicio	/sbin/reload	# <code>initctl reload ssh</code>

O bien:

```
# reload ssh
```

e. Activar/desactivar los servicios durante el arranque

Los archivos ubicados en la carpeta `/etc/init` permiten indicar si el servicio está activado o desactivado. Todos los archivos con extensión `.conf` están activos. Para desactivar el archivo podemos renombrarlo:

```
# mv /etc/init/ssh.conf /etc/init/ssh.conf.inactivo
```

Para reactivar el servicio, escribimos:

```
# mv /etc/init/ssh.conf.inactivo /etc/init/ssh.conf
```

3. Gestión de servicios con CentOS

Las versiones anteriores a la 6 emplean `sysinit`. La versión 6 utiliza `upstart`. La versión 7 emplea `systemd`.

a. Mostrar el estado de un servicio

Debemos utilizar el comando `systemctl`.

Sintaxis

```
systemctl <opción> <servicio>.service
```

Opción necesaria:

Opción	Descripción
estado	Muestra el estado del servicio.

Ejemplo de ejecución

```
# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since dim. 2015-10-04 17:44:59 CEST;
35s ago
     Process: 26665 ExecStartPre=/usr/sbin/sshd-keygen (code=exited,
status=0/SUCCESS)
    Main PID: 26666 (sshd)
      CGroup: /system.slice/sshd.service
              └─26666 /usr/sbin/sshd -D

oct. 04 17:44:59 rhsrv01.staff.local systemd[1]: Started OpenSSH
server daemon.
oct. 04 17:44:59 rhsrv01.staff.local sshd[26666]: Server listening
on 0.0.0.0 port 22.
oct. 04 17:44:59 rhsrv01.staff.local sshd[26666]: Server listening
on :: port 22.
```

b. Ver el estado de todos los servicios

Vamos a usar el comando `systemctl` con la opción `type` para especificar que se quiere ver los servicios.

```
# systemctl --type service
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
abrt-ccpp.service                  loaded active exited Install ABRT coredump hook
abrt-oops.service                  loaded active running ABRT kernel log watcher
abrt-xorg.service                  loaded active running ABRT Xorg log watcher
abrtd.service                      loaded active running ABRT Automated Bug
Reporting Tool
accounts-daemon.service            loaded active running Accounts Service
*** Salida truncada ***
```

c. Arrancar/detener un servicio

El comando `service` arranca, detiene o reinicia un demonio. Hace una llamada a `systemctl`.

Ejemplo

Detener el servicio ssh:

```
# service sshd stop
Redirecting to /bin/systemctl stop  sshd.service
```

Arrancar el servicio ssh:

```
# service sshd start
Redirecting to /bin/systemctl start  sshd.service
```

Reiniciar el servicio ssh:

```
# service sshd restart
Redirecting to /bin/systemctl restart  sshd.service
```

d. Recargar la configuración de un servicio

Podemos siempre usando el comando `service` recargar la configuración del servicio a condición de que el script se haga cargo de esta funcionalidad.

Ejemplo

Recargar ssh:

```
# Service sshd reload
Redirecting to /bin/systemctl reload sshd.service
```

e. Activar/desactivar los servicios durante el arranque

El comando chkconfig utilizada en las versiones anteriores de CentOS se substituye por systemctl.

Sintaxis

```
systemctl <opción> <nombre_del_servicio>
```

Opciones:

Opción	Descripción
enable	Activa el servicio al arrancar el sistema.
disable	Desactiva el servicio al arrancar el sistema.
is-enabled	Comprueba si el servicio está activo o no en el arranque del sistema.

Ejemplo

Desactivar el servicio ssh para todos los niveles de ejecución:

```
# systemctl disable sshd
rm '/etc/systemd/system/multi-user.target.wants/sshd.service'
```

Verificación:

```
# systemctl is-enabled sshd
disabled
```

Activar ssh:

```
# systemctl enable sshd
ln -s '/usr/lib/systemd/system/sshd.service'
'/etc/systemd/system/multi-user.target.wants/sshd.service'
```

Verificación:

```
# systemctl is-enabled sshd  
enabled
```

Gestores de arranque

Un gestor de arranque (boot loader) es un programa que permite cargar un sistema operativo a la vez. Puede ofrecernos un menú para seleccionar una opción del sistema a arrancar. En este caso, hablamos de multi-boot.

El firmware BIOS cargará el MBR (Master Boot Record), o sea, los 512 primeros bytes del disco. Luego, leerá el VBR (Volume Boot Record) de la partición y encontrará la ubicación del gestor de arranque.

En algunos equipos, no es el firmware BIOS sino el firmware EFI quien localiza el gestor de arranque.

Linux dispone de varios cargadores de arranque:

- Lilo
- GRUB legacy
- GRUB PC

1. LILO

LILO (LIinux LOader), desarrollado por John Coffman bajo licencia BSD, es el primer gestor de arranque del sistema GNU/Linux. Su configuración se encuentra en el archivo de texto `/etc/lilo.conf`. Ha ido dejando de forma progresiva su lugar a GRUB (GRand Unified Bootloader).

2. GRUB legacy

La versión 1 de GRUB, desarrollada bajo licencia GNU GPL versión 3, se llama GRUB legacy o GRUB 0.9x.

Gestiona otros sistemas como GNU/Linux o Microsoft Windows como Hurd, OpenBSD, FreeBSD, NetBSD, etc. Y soporta el multi-boot (arranque múltiple). Puede cargar una imagen de sistema operativo desde una red utilizando el protocolo TFTP.

De forma transparente, GRUB soporta varios sistemas de archivos como FFS (BSD), FAT16, FAT32, Minix fs, ext2, ReiserFS, JFS, XFS, etc. Puede a su vez descomprimir archivos de forma automática con gzip.

GRUB proporciona una interfaz con un menú que muestra los comandos de arranque predefinidos. Un tiempo de espera configurable nos permite intervenir, en caso contrario se

ejecuta la opción por defecto. Posee a su vez una interfaz por línea de comandos para cambiar los parámetros de arranque.

3. GRUB-PC

La versión 2, llamada GRUB-PC, ofrece dos evoluciones dignas de mención. Integra nuevos sistemas de archivo como ext4 o btrfs. Ofrece la posibilidad de escribir scripts con estructuras de control (bucles, condiciones...).

Las distribuciones Debian 8, Ubuntu Server 14.04 LTS y CentOS 7 utilizan GRUB 2.



El capítulo Secuencia de arranque aborda el mantenimiento de GRUB.

Registros del sistema

Los registros del sistema (logs) permiten seguir la actividad del sistema y se encuentran en la carpeta `/var/log`.

Los registros varían de una distribución a otra.

1. Registros de Debian

Los registros específicos de Debian:

Registro	Formato	Descripción
alternatives.log	texto	Información de las update-alternatives (actualizaciones alternativas). Las update-alternatives gestionan los enlaces simbólicos que determinan los comandos por defecto.
apt	carpeta	Información del comando apt.
auth.log	texto	Información de autorización del sistema, es decir, las conexiones de los usuarios y servicios.
btmpt	binario	Información de los intentos de conexión fallidos. Comando: last -f /var/log/btmpt less
cups	carpeta	Información sobre el demonio Common Unix Printing System.
daemon.log	texto	Información sobre los demonios del sistema.
debug	texto	Información sobre la depuración de todas las instalaciones.
dmesg	texto	Muestra el buffer de mensajes del kernel. Comando: dmesg less
dpkg.log	texto	Información sobre la instalación de paquetes .deb.
exim4	carpeta	Información del programa exim.
faillog	binario	Información de los intentos de conexión fallidos. Comando: faillog -a

fontconfig.log	texto	Información sobre la librería de software para X Window System que permite configurar las fuentes.
fsck	carpeta	Información del comando fsck.
hp	carpeta	Información sobre HP.
installer	carpeta	Información sobre la instalación del sistema.
kern.log	texto	Información del kernel de Linux.
lastlog	binario	Muestra información de las conexiones recientes para todos los usuarios. Comando: lastlog
mensajes	texto	Información general del sistema.
syslog	texto	Registro estándar del sistema.
user.log	texto	Información sobre todos los registros a nivel usuario.
wtmp	binario	Información sobre todas las conexiones e histórico de las desconexiones. Comando: last

2. Registros de Ubuntu

Registro	Formato	Descripción
alternatives.log	texto	Información de las update-alternatives (actualizaciones alternativas). Las update-alternatives gestionan los enlaces simbólicos que determinan los comandos por defecto.
apt	carpeta	Información del comando apt.
auth.log	texto	Información de autorización del sistema, es decir, las conexiones de los usuarios y servicios.
boot.log	texto	Información sobre el arranque del sistema.
btmpt	binario	Información de los intentos de conexión fallidos. Comando: <code>last -f /var/log/btmpt less</code>
dist-upgrade	carpeta	Información sobre la actualización del sistema.
dmesg	texto	Muestra el buffer de mensajes del kernel. Comando: <code>dmesg less</code>
dpkg.log	texto	Información sobre la instalación de paquetes .deb.
faillog	binario	Información de los intentos de conexión fallidos. Comando: <code>faillog -a</code>
fsck	carpeta	Información del comando fsck.
installer	carpeta	Información sobre la instalación del sistema.
kern.log	texto	Información del kernel de Linux.

landscape	carpeta	Información sobre los errores de la utilidad landscape-sysinfo.
lastlog	binario	Muestra información de las conexiones recientes para todos los usuarios. Comando: lastlog
syslog	texto	Registro estándar del sistema.
udev	texto	Información sobre los dispositivos.
unattended-upgrades	carpeta	Información sobre el proceso de actualización, en particular de seguridad.
upstart	carpeta	Información sobre los errores de upstart.
wtmp	binario	Información sobre todas las conexiones e histórico de las desconexiones. Comando: last

3. Registros de CentOS

En cuanto la instalación de CentOS concluye con éxito, los registros del programa de instalación anaconda se copian desde el directorio /tmp a /var/log.

Registro	Formato	Descripción
anaconda	carpeta	Información sobre la actualización del sistema.
audit	carpeta	Información del demonio Linux Audit (auditd).
boot.log	texto	Información sobre el arranque del sistema.
btmp	binario	Información de los intentos de conexión fallidos. Comando: last -f /var/log/btmp less
chrony	carpeta	Información sobre la aplicación Chrony que es un gestor de tiempo.
cron	texto	Información sobre el demonio de planificación de tareas cron.
cups	carpeta	Información sobre el demonio Common Unix Printing System.
dmesg	texto	Muestra el buffer de mensajes del kernel. Comando: dmesg less
gdm	carpeta	Información sobre el entorno GNOME.
lastlog	binario	Muestra información de las conexiones recientes para todos los usuarios. Comando: lastlog
libvirt	carpeta	Información sobre la virtualización.
maillog	texto	Información del servidor de correo.

mensajes	texto	Información general del sistema.
pluto	carpeta	Información sobre el demonio pluto.
pm-powersave.log	texto	Información sobre el gestor de alimentación.
qemu-ga	carpeta	Información sobre el software Qemu guest agent.
sa	carpeta	El comando sadc recopila datos sobre el uso del sistema y los graba, para un análisis posterior, en un archivo situado por defecto en este directorio.
samba	carpeta	Información sobre la aplicación Samba.
secure	texto	Información sobre la seguridad del sistema.
speech-dispatcher	carpeta	Información sobre la aplicación Speech Dispatcher.
spooler	texto	Información sobre los errores de correo y news.
tallylog	texto	Información sobre el módulo PAM (Pluggable Authentication Modules) de bloqueo de cuentas: pam_tally2.
tuned	carpeta	Información sobre el demonio tuned que supervisa el uso de los componentes del sistema.
wtmp	binario	Información sobre todas las conexiones y el histórico de las desconexiones. Comando: last
Xorg.0.log	texto	Información sobre Xorg.
yum.log	texto	Información del programa yum.

4. Leer los registros del sistema

Los registros en formato de texto están accesibles mediante los comandos que manipulan archivos. En cuanto a los registros que están en formato binario, el acceso se realiza obligatoriamente a través de un comando del shell dedicado a ellos.

Para las siguientes operaciones, utilizamos la distribución Debian. El planteamiento es idéntico con Ubuntu y CentOS.

a. Leer un registro texto

La lista de comandos para la lectura de archivos de texto es muy amplia. Estos son los principales comandos:

Comando	Descripción
cat	Muestra el contenido del archivo en la salida estándar (stdout).
more	Filtro que permite desplazarse a través de un texto pantalla a pantalla.
less	Filtro que permite desplazarse por un texto. less es un programa más avanzado que more.
head	Muestra el comienzo del archivo en n líneas.
tail	Muestra el final del archivo en n líneas.
grep	Muestra las líneas de un archivo que corresponden a un patrón específico.
awk	Lenguaje para manipular archivos en formato texto.

Leer todo el registro dmesg

El archivo `/var/log/dmesg` es demasiado largo para mostrarlo completo por pantalla. El comando preferido es less:

Sintaxis

```
less <nombre_de_registro>
```

Teclee en un terminal:

```
# less /var/log/dmesg
```


Y obtendremos en pantalla:

```
bob@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpuacct
[ 0.000000] Linux version 3.16.0-4-amd64 (debian-kernel@lists.debian.org) (gcc version 4.8.4 (Debian 4.8.4-1) ) #1 SMP Debian 3.16.7-ckt25-2 (2016-04-08)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.16.0-4-amd64 root=/dev/mapper/SystemVG-rootLV ro quiet
[ 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 0x0000000000100000-0x00000000007ffeffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000007fff0000-0x00000000007fffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000fffc0000-0x0000000000ffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 0.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[ 0.000000] AGP: No AGP bridge found
[ 0.000000] e820: last_pfn = 0x7fff0 max_arch_pfn = 0x400000000
[ 0.000000] MTRR default type: uncachable
```

Con las flechas de dirección es posible navegar en el texto. La letra q nos saca de less.

Mostrar las 10 primeras líneas del diario dmesg

Seleccionamos el comando head.

Sintaxis

```
head <número_de_líneas_a_mostrar> <nombre_del_archivo>
```

Teclee en un terminal:

```
# head -10 /var/log/dmesg
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] 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 3.2.41-2+deb7u2
[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.2.0-4-amd64 root=/dev/mapper/rootvg-root ro quiet
[ 0.000000] BIOS-provided physical RAM map:
```

```
[ 0.000000] BIOS-e820: 0000000000000000 - 000000000009fc00
(usable)
[ 0.000000] BIOS-e820: 000000000009fc00 - 00000000000a0000
(reserved)
[ 0.000000] BIOS-e820: 00000000000f0000 - 0000000000100000
(reserved)
[ 0.000000] BIOS-e820: 0000000000100000 - 000000001ffff000
(usable)
[ 0.000000] BIOS-e820: 000000001ffff000 - 0000000020000000
(ACPI data)
```

Mostrar las 8 primeras líneas del registro dmesg

Elegimos el comando tail.

Sintaxis

```
tail <número_de_líneas_a_mostrar> <nombre_del_archivo>
```

Teclee en un terminal:

```
# tail -8 /var/log/dmesg
[ 8.136024] RPC: Registered tcp transport module.
[ 8.136025] RPC: Registered tcp NFSv4.1 backchannel transport
module.
[ 8.168233] FS-Cache: Loaded
[ 8.213843] FS-Cache: Netfs 'nfs' registered for caching
[ 8.228745] Installing knfsd (copyright (C) 1996
okir@monad.swb.de).
[ 9.175446] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 9.177264] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex,
Flow Control: RX
[ 9.177788] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Mostrar las líneas que contengan la palabra clave "eth0"

Seleccionamos el comando grep.

Sintaxis

```
grep <patrón> <nombre_del_archivo>
```

Teclee en un terminal:

```
# grep eth0 /var/log/dmesg
[ 3.378150] e1000 0000:00:03.0: eth0: (PCI:33MHz:32-bit)
08:00:27:bb:0c:81
[ 3.378156] e1000 0000:00:03.0: eth0: Intel(R) PRO/1000 Network
Connection
[ 9.175446] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 9.177264] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex,
Flow Control: RX
[ 9.177788] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

b. Leer un registro binario

El uso de algunos comandos del shell permite acceder a los archivos de registro binarios como last, lasb, lastlog, who...

Saber quién está conectado al equipo

El comando who leerá del archivo /var/log/wtmp.

```
# who
bob      tty1          2013-07-13 14:22
bob      pts/0         2013-07-13 14:33 (jersey.local)
```

c. Logwatch

El programa logwatch permite controlar los registros del sistema y crear informes. No se instala por defecto, pero se encuentra en los repositorios.

Instalación en Debian y Ubuntu

```
# apt-get install logwatch
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho

Se instalarán los siguientes paquetes extras:
```

```
libdate-manip-perl libsys-cpu-perl
Paquetes sugeridos:
fortune-mod
Se instalarán los siguientes paquetes NUEVOS:
libdate-manip-perl libsys-cpu-perl logwatch
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 1.299 kB de archivos.
Se utilizarán 13,2 MB de espacio de disco adicional después de esta
operación.
*** Salida truncada ***
```

Instalación en CentOS

```
# yum install logwatch
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror0.babylon.network
* extras: mirror0.babylon.network
* updates: fr2.rpmfind.net
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete logwatch.noarch 0:7.4.0-28.20130522svn140.el7 debe
ser instalado
--> Procesando dependencias: perl(Sys::MemInfo) para el paquete:
logwatch-7.4.0-28.20130522svn140.el7.noarch
--> Procesando dependencias: perl(Sys::CPU) para el paquete:
logwatch-7.4.0-28.20130522svn140.el7.noarch
--> Procesando dependencias: perl(Date::Manip) para el paquete:
logwatch-7.4.0-28.20130522svn140.el7.noarch
--> Ejecutando prueba de transacción
---> Paquete perl-Date-Manip.noarch 0:6.41-2.el7 debe ser instalado
---> Paquete perl-Sys-CPU.x86_64 0:0.54-4.el7 debe ser instalado
---> Paquete perl-Sys-MemInfo.x86_64 0:0.91-7.el7 debe ser instalado
--> Resolución de dependencias finalizada
```

```
*** Salida truncada ***
```

Si deseamos cambiar la configuración, editamos el archivo `/usr/share/logwatch/default.conf/logwatch.conf`, es decir, la dirección de destino de los informes (MailTo) o el nivel de detalle (Detail).

Modificar el nivel de detalle

Abrimos el archivo de configuración con nano:

```
# nano /usr/share/logwatch/default.conf/logwatch.conf
```

Buscamos la línea Detail y cambiamos el valor low por High:

```
Detail = High
```

Guardamos con la combinación de teclas CTRL + O y salimos con CTRL + X.

Prueba del funcionamiento de logwatch

Teclee en un terminal:

```
# logwatch
```

Obtenemos un resultado como este:

```
##### Logwatch 7.4.0 (25/04/16) #####
```

```
Processing Initiated: Mon Jul 25 19:29:49 2016
```

```
Date Range Processed: yesterday
```

```
( 2016-Jul-24 )
```

```
Period is day.
```

```
Detail Level of Output: 10
```

```
Type of Output/Format: stdout / text
```

```
Logfiles for Host: Debian-pc
```

```
#####
```

```
----- Cron Begin -----
```

```
Commands Run:
```

```
User root:
```

```
cd / && run-parts --report /etc/cron.hourly: 10 Time(s)
```

----- Cron End -----

----- EXIM Begin -----

--- Queue Runners ---

Start queue run: 18 Time(s)

End queue run: 18 Time(s)

----- EXIM End -----

...

----- pam_unix Begin -----

cron:

Sessions Opened:

root: 10 Time(s)

----- pam_unix End -----

----- Disk Space Begin -----

Filesystem	Size	Used	Avail	Use%	Mounted on
rootfs	322M	149M	157M	49%	/
udev	10M	0	10M	0%	/dev
/dev/mapper/rootvg-root	322M	149M	157M	49%	/
/dev/sda1	228M	9.9M	206M	5%	/boot
/dev/mapper/rootvg-home	3.5G	72M	3.3G	3%	/home
/dev/mapper/rootvg-tmp	291M	11M	266M	4%	/tmp
/dev/mapper/rootvg-usr	3.4G	997M	2.3G	31%	/usr
/dev/mapper/rootvg-var	1.7G	251M	1.4G	16%	/var

```
----- Disk Space End -----
```

```
##### Logwatch End #####
```

Planificación de logwatch

Tras la instalación de logwatch, se configura una tarea diaria en el archivo `/etc/cron.daily/0logwatch`:

```
# cat /etc/cron.daily/0logwatch
```

Obtenemos:

```
#!/bin/sh

#Set logwatch location
LOGWATCH_SCRIPT="/usr/sbin/logwatch"

#Add options to this line. Most options should be defined in /etc/
logwatch/conf/logwatch.conf,
#but some are only for the nightly cronrun such as --output mail
and should be set here.
#Other options to consider might be "--format html" or "--encode
base64", man logwatch for more details.
OPTIONS="--output mail"

#Call logwatch
$LOGWATCH_SCRIPT $OPTIONS

exit 0
```

5. rsyslog

En las distribuciones Debian, Ubuntu y CentOS, el software syslog se sustituye por rsyslog. Aporta algunas mejoras, incluidos eventos que pueden escribirse en una base de datos (por ejemplo MariaDB o PostgreSQL), el uso de su propio protocolo PHWR (Reliable Event Logging Protocol) en lugar de UDP (User Datagram Protocol), etc., sin embargo mantiene las reglas de syslog.

El programa que se ejecuta como un demonio en el sistema es `/usr/sbin/rsyslogd`.

El archivo de configuración de rsyslog se llama `/etc/rsyslog.conf`. Una carpeta `/etc/rsyslog.d` permite incluir archivos de configuración adicionales cuya extensión es `.conf`. Esto se define mediante la siguiente línea en el archivo `/etc/rsyslog.conf`:

```
# grep IncludeConfig /etc/rsyslog.conf
$IncludeConfig /etc/rsyslog.d/*.conf
```

a. rsyslog.conf

El archivo `rsyslog.conf` se divide en tres partes:

- Los módulos a cargar.
- Las directrices globales como los permisos por defecto para un archivo de registro o la ubicación de los archivos de configuración adicionales.
- Las normas para definir los registros y sus contenidos. Las reglas pueden estar escritas en los archivos `.conf` almacenados en la carpeta `/etc/rsyslog.d`.

b. Las reglas de rsyslog

Si observamos las siguientes líneas procedentes del archivo `/etc/rsyslog.conf` de Debian:

<code>mail.info</code>	<code>-/var/log/mail.info</code>
<code>mail.warn</code>	<code>-/var/log/mail.warn</code>
<code>mail.err</code>	<code>/var/log/mail.err</code>

Apreciamos que incluye dos columnas:

- el origen del mensaje,
- el destino del mensaje

El origen también se compone de dos partes:

- las facilidades que definen los tipos de mensajes,
- el nivel de gravedad.

Disponemos de 24 facilidades (facilities) numeradas del 0 al 23:

Nº	Palabra clave	Descripción
0	kern	Mensaje del kernel
1	user	Mensajes de nivel usuario
2	mail	Mensajes de sistema
3	daemon	Demonios del sistema
4	auth	Mensaje de autorización
5	syslog	Mensajes generados por rsyslogd
6	lpr	Subsistemas de impresora
7	news	Subsistemas de noticias
8	uucp	Subsistemas uucp
9	- -	Daemon clock (reloj del sistema)
10	authpriv	Mensajes de autorización
11	ftp	Demonio FTP
12	- -	Subsistemas NTP
13	- -	Audit
14	- -	Alerta
15	cron	Demonio cron
16	local0	Local0

17	local1	Local1
18	local2	Local2
19	local3	Local3
20	local4	Local4
21	local5	Local5
22	local6	Local6
23	local7	Local7

Los niveles de gravedad (Security level) son, por su parte, ocho numerados de 0 a 7.

Nº	Palabra clave	Descripción
0	Emerg	Emergencia. Sistema inutilizable.
1	Alert	Alerta. Intervención inmediata necesaria.
2	Crit	Error de sistema crítico.
3	Err	Error de funcionamiento.
4	Warning	Advertencia.
5	Notice	Evento normal que debe ser señalado.
6	Info	Para información.
7	debug	Mensaje de depuración.

Si volvemos a observar las siguientes líneas:

```
mail.info          -/var/log/mail.info
mail.warn          -/var/log/mail.warn
mail.err           /var/log/mail.err
```

Analizamos la primera línea. La facilidad mail y el nivel de gravedad info se envían al registro /var/log/mail.info.

Un guión delante de la ruta mejora el rendimiento de escritura, ya que no hay una sincronización del archivo. Es útil para los archivos de registro que almacenan información poco crítica. Sin embargo, podemos perder datos durante un crash del sistema operativo.

6. Logrotate

El programa logrotate se encarga de efectuar el rotado de los registros del sistema operativo.

El programa cron ejecuta el archivo /etc/cron.daily/logrotate todos los días.

```
# cat /etc/cron.daily/logrotate
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

El archivo de configuración de logrotate es /etc/logrotate.conf:

```
# cat logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
```

```
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

El rotado de los registros wtmp y btmp se realiza aquí. La directiva include permite incluir en la carpeta /etc/logrotate.d archivos de configuración para el rotado de los registros:

```
# ls -l /etc/logrotate.d
total 28
-rw-r--r-- 1 root root 173 abr 13 18:52 apt
-rw-r--r-- 1 root root 79 nov 8 2014 aptitude
-rw-r--r-- 1 root root 181 jun 9 10:42 cups-daemon
-rw-r--r-- 1 root root 232 abr 10 05:50 dpkg
-rw-r--r-- 1 root root 146 feb 17 2015 exim4-base
-rw-r--r-- 1 root root 126 feb 17 2015 exim4-paniclog
```

```
-rw-r--r-- 1 root root 515 oct  2 2014  rsyslog
```

Ejemplo: Contenido del archivo cups-daemon:

El archivo /etc/logrotate.d/cups-daemon configura el rotado del registro /var/log/cups del programa CUPS (Common UNIX Printing System):

```
# cat /etc/logrotate.d/cups-daemon
/var/log/cups/*log {
    daily
    missingok
    rotate 7
    sharedscripts
    postrotate
        invoke-rc.d --quiet cups restart > /dev/null
    endscript
    compress
    delaycompress
    notifempty
    create
}
```

Las principales directivas que podemos invocar son:

Directiva	Descripción
daily weekly monthly	Define el período: todos los días, semanas o bien todos los meses.
missingok	Si el archivo de registro no se encuentra se considera normal.
rotate <valor>	Establece el número de archivos a conservar.
sharedscripts	Ejecuta una sola vez por rotado el script mencionado en la directiva postrotate/endscript.
postrotate endscript	Las líneas situadas entre ambas directivas se ejecutan antes de cambiar el registro.
compress	Comprime los archivos guardados.
notifempty	No efectúa el rotado del registro cuando está vacío.
create <permisos> <propietarios>	Los archivos guardados poseen los permisos, la cuenta y el grupo de propietarios mencionados.

Aplicaciones

Los programas binarios están escritos en un lenguaje de programación C, C++, Java, Mono (implementación de la plataforma de desarrollo de Microsoft .NET en open source), Gambas, FreePascal, etc.

Los programas interpretados están escritos en TCL/TK, Perl, Python, Ruby, etc.

Ya hemos hablado de las librerías GTK+ y Qt para el desarrollo de una aplicación gráfica. Las aplicaciones de consola podrán usar la librería ncurses (New Curses) que ofrece de forma principal la gestión de ventanas, ocho colores diferentes, la gestión del ratón y la gestión de las teclas de función del teclado. Por supuesto, existen otras librerías.

Los programas binarios son de 32 o 64 bits. El comando file permite identificar su modo de compilación.

Sintaxis

```
file <ruta/nombre_del_archivo>
```

Ejemplo

Conocer si el programa ls es de 32 o 64 bits.

Teclee en un terminal:

```
# file /bin/ls
/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.32,
BuildID[sha1]=aa7ff68f13de25936a098016243ce57c3c982e06, stripped
```

Aquí vemos 64-bit en el resultado.

1. Procesos

a. Definición

Un proceso es un conjunto de instrucciones a ejecutar en una dirección de memoria que le pertenece de 4 GB de RAM para un procesador de 32 bits y 128 TB para un procesador de 64 bits.

El kernel pondrá a su disposición esta memoria virtual:

- Asociándole RAM real bajo demanda (paging).
- Mapeando una parte del disco bajo demanda (map).
- Almacenando el contenido en la partición swap.

La memoria asignada al proceso contiene:

- 1 GB reservado para el kernel.
- El código que corresponde a las instrucciones a ejecutar.
- Las librerías. El código invoca a las librerías de funciones.
- La pila. Es el lugar donde se almacenan las variables locales y las direcciones de retorno de las funciones.
- El proceso, que es el resto de la memoria asignada por tramos.

Un proceso puede ser iniciado por:

- un usuario,
- a través de un dispositivo,
- otro proceso.

b. Identificador de proceso

El sistema asignará un PID (Process IDentifier) a los procesos durante su tiempo de ejecución. Poseen también el PID de su respectivo padre: PPID (Parent Process IDentifier).

El primer programa arrancado por el sistema se llama init y posee el PID 1. Es el único proceso que no tiene padre. Se inicia durante el arranque del sistema por el kernel de Linux.

Los procesos siguientes incrementan su número hasta el 32768. Este valor por defecto puede configurarse en el archivo `/proc/sys/kernel/pid_max`. Cuando el valor máximo se alcanza, la asignación de los PID vuelve a 2, evitando los PID ya asignados.

c. Estado de un proceso

Durante el período de ejecución de un proceso, este puede pasar por diferentes estados:

- Activo. El proceso hace uso del procesador para realizar las operaciones necesarias para su ejecución.
- Ejecutable. El proceso espera a que el proceso activo le ceda su lugar.
- Dormido. El proceso espera una condición para volver a ser ejecutable o activo.
- Zombie. El proceso en realidad ha terminado y no debería existir. Sin embargo, por diversas razones, el proceso padre no ha sido informado de este hecho. El sistema conserva mientras la información correspondiente al proceso hijo para que su padre pueda ver que ha terminado su ejecución.



El usuario no ve la diferencia entre los dos estados activo y ejecutable. Ve un programa en ejecución. Este matiz se encuentra a nivel del kernel.

d. Función fork()

Cuando un proceso llama a la función fork(), se duplica a sí mismo.

Ejemplo escrito en lenguaje C

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(void)
{
    pid_t pid = fork();

    if(pid == 0)
    {
        printf("Hijos: PID = %d - PPID = %d\n", getpid(), getppid());
    }
    else if(pid != -1)
    {
        printf("Padre: PID = %d - PPID = %d\n", getpid(), getppid());
    }
    else
    {
        perror("error");
        return 1;
    }

    return 0;
}
```

Introducimos el código con un editor de texto y lo guardamos con el nombre progfork.c. Luego, lo compilamos con gcc (GNU Compiler Collection):

```
# gcc -o progfork progfork.c
```

Para ejecutar el programa:

```
# ./progfork
```

Este es el resultado:

```
Padre: PID = 2971 - PPID = 2925
Hijos: PID = 2972 - PPID = 2971
```

Por supuesto, los valores de los PID y PPID cambian con cada ejecución.

e. Función exec()

Un programa puede sustituirse a sí mismo usando la función exec().

Ejemplo escrito en lenguaje C

El programa progexecl será sustituido por el comando ls -rtl:

```
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    execl("/bin/ls", "/bin/ls", "-rtl", (char *) 0);
    perror("execl");
    return 1;
}
```

Introducimos el código con un editor de texto y lo guardamos con el nombre progexecl.c. Luego, lo compilamos con gcc:

```
# gcc -o progexecl progexecl.c
```

La ejecución del programa se realiza así:

```
# ./progexecl
```

El resultado demuestra que el programa progexec1.c se substituye con el programa /bin/ls -rtl:

```
total 16
-rw-r--r-- 1 root root 138 abr 27 15:53 progexec1.c
-rwxr-xr-x 1 root root 8560 abr 27 15:53 progexec1
```

2. Threads

Una aplicación puede contener uno o varios hilos de ejecución (threads) que se ejecutan de forma casi simultánea o simultánea en los procesadores multicore.

Los threads comparten el mismo espacio de memoria protegida y los descriptores de archivos. Lo que diferencia un thread de otro, son los punteros de ejecución y de pila. Dos threads pueden ejecutar extremos de código distintos de un mismo ejecutable y tener variables locales diferentes de los demás threads del proceso.

Para crear un nuevo thread, el proceso llama a la función pthread_create() que duplica la pila existente, así como el estado del procesador. El planificador del kernel verá entonces una nueva entrada.

Ejemplo escrito en lenguaje C:

```
//progthreads.c

#include <stdio.h>
#include <stdlib.h>
#include </usr/include/pthread.h>

void *mostrar_mensaje( void *ptr );

int main(void)
{
    pthread_t thread1, thread2;
    char *mensaje1 = "Thread 1";
    char *mensaje2 = "Thread 2";
    int iret1, iret2;

    iret1 = pthread_create( &thread1, NULL, mostrar_mensaje,
```

```

(void*) mensaje1);
    iret2 = pthread_create( &thread2, NULL, mostrar_mensaje,
(void*) mensaje2);

    pthread_join( thread1, NULL);
    pthread_join( thread2, NULL);

    printf("Thread 1 devuelve : %d\n",iret1);
    printf("Thread 2 devuelve : %d\n",iret2);
    return 0;
}

void *mostrar_mensaje( void *ptr )
{
    char *mensaje;
    mensaje = (char *) ptr;
    printf("%s \n", mensaje);
}

```

Escribimos el código con un editor de texto y lo guardamos con el nombre progthreads.c. Luego, lo compilamos con gcc:

```
# gcc -o progthreads progthreads.c -lpthread
```

Para ejecutar el programa, escribimos en un terminal:

```
# ./progthreads
```

Este es el resultado:

```

Thread 1
Thread 2
Thread 1 devuelve: 0
Thread 2 devuelve: 0

```



Para profundizar en el tema, consulte el capítulo Mantenimiento de las aplicaciones.

Distribuciones

Como recordamos, una distribución GNU/Linux es un sistema operativo GNU basado en el kernel de Linux. Cada una de estas está desarrollada bajo unos objetivos y una filosofía que le son propias. Para alguien que descubre Linux, esta elección de versiones de Linux nos puede perturbar un poco.

1. Identificar la distribución

Podemos agruparlas éstas por familia cuya identidad sería el resultado de una distribución «raíz»: Debian, Red Hat, SLS, enoch, arch...

Luego tenemos las distribuciones derivadas (fork), o derivadas de una distribución derivada y así sucesivamente...

He aquí algunos ejemplos:

Distribución	Distribución derivada
Debian SID	Ubuntu Desktop o Server
Ubuntu Desktop	Linux Mint
Debian SID	Linux Mint Debian Edition (LMDE)
Fedora	Red Hat Enterprise Linux
Red Hat Enterprise Linux	CentOS

2. Debian

a. Presentación

Debian es una distribución de software libre con fines no comerciales, creada por Ian Murdock en 1993. Su organización se basa en una comunidad democrática.

b. Diferentes versiones de Debian

Debian se compone de tres ramas de desarrollo:

- unstable
- testing
- stable

Unstable

Debian SID (Still in Development) corresponde siempre a la versión inestable (unstable). Está en constante evolución (rolling release).

No se recomienda su uso en producción. Es una pre-versión.

Testing

Debian Testing es la futura versión estable, incluyendo paquetes con un grado suficiente de estabilidad.

Stable

La versión de Debian estable tiene un número de versión y un nombre de personaje de la película de animación Toy Story de Pixar:

- Debian 6 Squeeze
- Debian 7 Wheezy
- Debian 8 Jessie
- Debian 9 Stretch será la próxima versión.

La versión estable solo se actualiza para la corrección de errores o problemas de seguridad. Es la versión que hay que utilizar en producción.

Sin embargo, los repositorios Backports oficialmente soportados por Debian ofrecen versiones de programas más recientes, pero con el riesgo de que sean menos estables.

3. Ubuntu

a. Presentación

Canonical Ltd es una empresa fundada por Mark Shuttleworth en 2004 que desea poner a disposición de todos la distribución GNU/Linux Ubuntu pero ofreciendo un soporte técnico y comercial.

b. Diferentes versiones de Ubuntu

Ubuntu, basado en la distribución Debian, ofrece una versión servidor y una versión estación de trabajo cada 6 meses. Se publican en abril y octubre de cada año, de ahí la numeración de versiones: año.mes. Por ejemplo: 12.04 para abril de 2012.

Cada dos años, en abril, se publica una versión LTS (Long-Term Support). Es la versión que hay que poner en producción ya que se garantiza un soporte durante cinco años.

- durante los dos primeros años, las correcciones de errores, actualizaciones de seguridad y soporte de nuevo hardware están garantizadas.
- durante los siguientes tres años, se publican correcciones de errores y actualizaciones de seguridad.

4. Red Hat

a. Presentación

Red Hat es una empresa situada en Raleigh (Carolina del Norte, USA) fundada en 1993 que desarrolla sistemas operativos y software open source.

b. Diferentes versiones de Red Hat

En sus orígenes, Red Hat basó su distribución comercial RHEL sobre RHL (Red Hat Linux) que concluyó con la versión 9 en 2003.

Derivada de RHL, el proyecto Fedora con apoyo de la sociedad Red Hat desarrolla la distribución Fedora Core (FC). Es una de las partes más importantes de RHEL de nuestros días.

En este cuadro encontrará la concordancia entre las versiones de la comunidad y las comerciales:

Versión de la comunidad	Versión comercial
Red Hat Linux 6.2	Red Hat Enterprise Linux
Red Hat Linux 7.2	Red Hat Enterprise Linux 2.1
Red Hat Linux 9.0	Red Hat Enterprise Linux 3
Fedora Core 3	Red Hat Enterprise Linux 4
Fedora Core 6	Red Hat Enterprise Linux 5
Fedora 12	Red Hat Enterprise Linux 6
Fedora 19	Red Hat Enterprise Linux 7

Si necesita usar un paquete RPM de repositorios externos a los de Red Hat, obtenga la versión correcta del sistema.

- el6 o fc12 para RHEL 6.
- el7 o fc19 para RHEL 7.

Tenga especial cuidado también con las arquitecturas i386, i586, i686, x86_64...

He aquí algunos motores de búsqueda de paquetes RPM.

Nombre	URL
Pbone	http://rpm.pbone.net
Rpmfind	http://rpmfind.net/

5. CentOS

La Organización The CentOS Project (Community ENTERprise Operating System) crea una distribución Linux que se distribuye de forma gratuita. Se basa en las fuentes de los paquetes de la distribución Red Hat Enterprise Linux. Por supuesto, los logotipos de Red Hat se sustituyen.

Podemos afirmar que el binario del sistema CentOS 6.4 es casi idéntico al de Red Hat Enterprise Linux.

Desde enero de 2014, Red Hat y CentOS se han aproximado. Los principales miembros de la Comunidad CentOS trabajan ahora en Red Hat. Se podría considerar que CentOS es una versión gratuita de Red Hat.

6. Identificar la versión de la distribución

Si cada programa que constituye el sistema operativo cuenta con su respectivo número de versión, la distribución tiene también una versión, como hemos podido comprobar.

a. Versión de Debian

Escribimos:

```
# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 8.2 (jessie)
Release:        8.2
Codename:       jessie
```

O leyendo el archivo `/etc/os-release` en el directorio `/etc`:

```
# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 8 (jessie)"
NAME="Debian GNU/Linux"
VERSION_ID="8"
VERSION="8 (jessie)"
ID=debian
HOME_URL="http://www.debian.org/"
SUPPORT_URL="http://www.debian.org/support/"
BUG_REPORT_URL="https://bugs.debian.org/"
```

b. Versión de Ubuntu

Escriba:

```
# lsb_release -a
```

```
No LSB modules are available.
```

```
Distributor ID: Ubuntu
```

```
Description:    Ubuntu 14.04.3 LTS
```

```
Release:        14.04
```

```
Codename:       trusty
```

Ubuntu proporciona dos archivos:

- /etc/lsb-release:

```
# cat /etc/lsb-release
```

```
DISTRIB_ID=Ubuntu
```

```
DISTRIB_RELEASE=12.04
```

```
DISTRIB_CODENAME=precise
```

```
DISTRIB_DESCRIPTION="Ubuntu 12.04.2 LTS"
```

- y /etc/os-release:

```
#cat /etc/os-release
```

```
NAME="Ubuntu"
```

```
VERSION="14.04.3 LTS, Trusty Tahr"
```

```
ID=ubuntu
```

```
ID_LIKE=debian
```

```
PRETTY_NAME="Ubuntu 14.04.3 LTS"
```

```
VERSION_ID="14.04"
```

```
HOME_URL="http://www.ubuntu.com/"
```

```
SUPPORT_URL="http://help.ubuntu.com/"
```

```
BUG_REPORT_URL=http://bugs.launchpad.net/ubuntu/
```

c. Versión de CentOS

El comando `lsb_release` no se instala por defecto en la versión 7.

```
# lsb_release -a
bash: lsb_release: no se encontró la orden....
```

Instale el paquete `redhat-lsb` con el comando `yum`.

```
# yum -y install redhat-lsb
Complementos cargados: fastestmirror, langpacks
base | 3.6 kB 00:00
extras | 3.4 kB 00:00
updates | 3.4 kB 00:00
Loading mirror speeds from cached hostfile
* base: mirror0.babylon.network
* extras: mirror0.babylon.network
* updates: fr2.rpmfind.net
Resolviendo dependencias
--> Ejecutando prueba de transacción
*** Salida truncada ***
```

Ahora que hemos añadido el comando, escribimos:

```
# lsb_release -a
LSB Version: :core-4.1-amd64:core-4.1-noarch:cxx-4.1-
amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-noarch:languages-4.1-
amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-noarch
Distributor ID: CentOS
Description: CentOS Linux release 7.2.1511 (Core)
Release: 7.2.1511
Codename: Core
```

También podemos leer el archivo `/etc/redhat-release` para conocer la versión de la distribución.

```
# cat /etc/redhat-release
CentOS Linux release 7.2.1511 (Core)
```

También está disponible el archivo `/etc/os-release`:

```
# cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
```

Plantilla del sistema

Para concluir, algo importante por hacer, que a veces se descuida por parte de los administradores, es construir una plantilla del sistema a mantener. Esto puede hacerse a mano, en papel o con ayuda de un software. Cuanto más preciso, más sencillas serán las intervenciones (solución de problemas, optimización, análisis del rendimiento, etc.).

Ejercicio

El objetivo es identificar los diferentes componentes del sistema.

1. Antes de comenzar, abra una consola en una de las tres distribuciones que comentamos en este libro. Luego, identifíquese como root.

2. Identifique, empleando diferentes comandos, los siguientes datos:

- La arquitectura física con `uname`,
- La versión del kernel,
- El nombre y la versión del shell,
- La versión de GRUB,
- La versión de la distribución de Linux,
- La lista de servicios arrancados.

Introducción

El capítulo Arquitectura del sistema GNU/Linux explica el cometido del kernel de Linux en el sistema, así como sus características.

El objetivo de este capítulo es abordar el diseño de un kernel adaptado a nuestras necesidades.

1. ¿Por qué compilar un kernel?

Ante todo, la primera pregunta que hay que plantearse: «¿Es realmente necesario y útil?».

Estas son las principales razones que nos pueden llevar a compilar un kernel:

- Comprender el funcionamiento del kernel de Linux.
- Hacer funcionar un equipo que no está soportado por el kernel actual.
- Aplicar uno o varios parches.
- Optimizar el núcleo por razones de rendimiento o seguridad.

Cabe señalar que se encuentra solo ante el ordenador mientras que un fabricante de software cuenta con un equipo estructurado para desarrollar un kernel. Por lo tanto, en la mayoría de los casos, el kernel que proviene de una distribución debería bastar para sus necesidades. Aunque esta no sea, a menudo, la última versión estable. En producción, es preferible adoptar una versión estable probada y comprobada en lugar de correr sin cesar tras la última versión. Esta última puede no contar con el soporte del fabricante.

2. Archivos que componen el núcleo

El núcleo está compuesto por tres partes:

- vmlinuz es la imagen del kernel.
- system.map contiene los símbolos del kernel requeridos por los módulos para garantizar el correcto arranque de las funciones del kernel.
- initrd (initial ram disk) carga los controladores compilados en los módulos necesarios para arrancar el kernel.

Estos tres archivos son dependientes y cada uno se regenera en cada compilación.

Compilar e instalar un kernel

Disponemos de dos métodos:

- La compilación clásica.
- La compilación al estilo Debian.

1. Realizar una compilación clásica

Este método puede utilizarse en todas las distribuciones GNU/Linux, incluyendo Debian.

a. Requisitos previos

Para Debian y Ubuntu Server, debemos instalar los paquetes necesarios para la compilación:

- build-essential
- module-init-tools
- fakeroot

En cuanto a CentOS, los paquetes a añadir son:

- Las herramientas de desarrollo (Development Tools), y las librerías de desarrollo (Development Libraries).
- La librería ncurses (ncurses-devel) para menuconfig o nconfig.
- La librería qt-devel para xconfig.
- Las librerías gtk2-devel, libglade2 para gconfig.

Se pueden añadir otros paquetes a la lista anterior en función de las necesidades o las dependencias.

Dentro del sistema, la ubicación para generar el kernel debe disponer de espacio en disco suficiente. Necesitará al menos 10 GB.

b. Descargar el código fuente del kernel

La descarga del código fuente del kernel se realiza desde el sitio web <http://www.kernel.org>. El comando `wget` permite realizar esta operación por línea de comandos. También puede usar un navegador web para efectuar esta operación.

La versión estable del kernel en el momento de escribir este libro es la 4.3. El nombre del archivo es `linux-4.3.tar.xz`.

Sintaxis

```
wget <URL>
```


Teclee en un terminal:

```
# wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.3.tar.xz
--2015-11-24 15:00:16-- https://cdn.kernel.org/pub/linux/kernel/v4.x/
linux-4.3.tar.xz
Resolviendo cdn.kernel.org (cdn.kernel.org)... 185.31.17.69
Conectando con cdn.kernel.org (cdn.kernel.org)|185.31.17.69|:443...
conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Tamaño: Longitud: 86920812 (83M) [application/x-xz]
Grabando a: « linux-4.3.tar.xz »

linux-4.3.tar.xz          100%[=====>]  82,89M  3,39MB/s
ds 24s

2015-11-24 15:01:17 (1,36 MB/s) - « linux-4.3.tar.xz » guardado
[86920812/86920812]
```

Para descomprimir el archivo hay que utilizar la herramienta tar.

Sintaxis

```
tar <opciones> <archivo_a_descomprimir>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-x	--extract	Extrae los archivos contenidos en el archivo.
-J	--xz	Descompresión en formato Lzma.
-v	--verbose	Muestra la lista detallada de los archivos tratados.
-f	--file	Utiliza el archivo comprimido especificado.

Escriba en un terminal:

```
# tar xJvf linux-4.3.tar.xz
linux-4.3/
linux-4.3/.get_maintainer.ignore
linux-4.3/.gitignore
linux-4.3/.mailmap
linux-4.3/COPYING
linux-4.3/CREDITS
linux-4.3/Documentation/
linux-4.3/Documentation/00-INDEX
linux-4.3/Documentation/ABI/
linux-4.3/Documentation/ABI/README
linux-4.3/Documentation/ABI/obsolete/
*** Salida truncada ***
```

La extracción se efectúa en la carpeta linux-4.3. Empleando el comando du, podemos conocer su tamaño.

Sintaxis

```
du <opciones> <carpeta>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-h	--human-readable	Muestra los tamaños en un formato legible por un ser humano.
-d	--max-depth	Muestra el espacio total ocupado por una carpeta solo si se encuentra a una profundidad inferior o igual al nivel indicado. --max-depth=0 es idéntico a --summarize.
-s	--summarize	Muestra solamente un total para cada parámetro.

Teclee en un terminal:

```
# du -h --max-depth=0 ./linux-4.3  
702M    ./linux-4.3
```

O bien:

```
# du -hs ./linux-4.3  
702M    ./linux-4.3
```

El tamaño en disco del código fuente del kernel Linux versión 4.3 es 702 MB.

c. Herramientas para configurar el kernel

Configurar adecuadamente todas las opciones del kernel Linux implica por una parte un buen conocimiento sobre el funcionamiento de Linux y del hardware utilizado (dispositivos, procesador(es), memoria, discos físicos y lógicos...) en el que se va a instalar el nuevo kernel de Linux. Pero también, por otra parte, hay que identificar el objetivo del diseño de este último.

Para efectuar la configuración del kernel, disponemos de cinco herramientas CLI o GUI:

`make config`

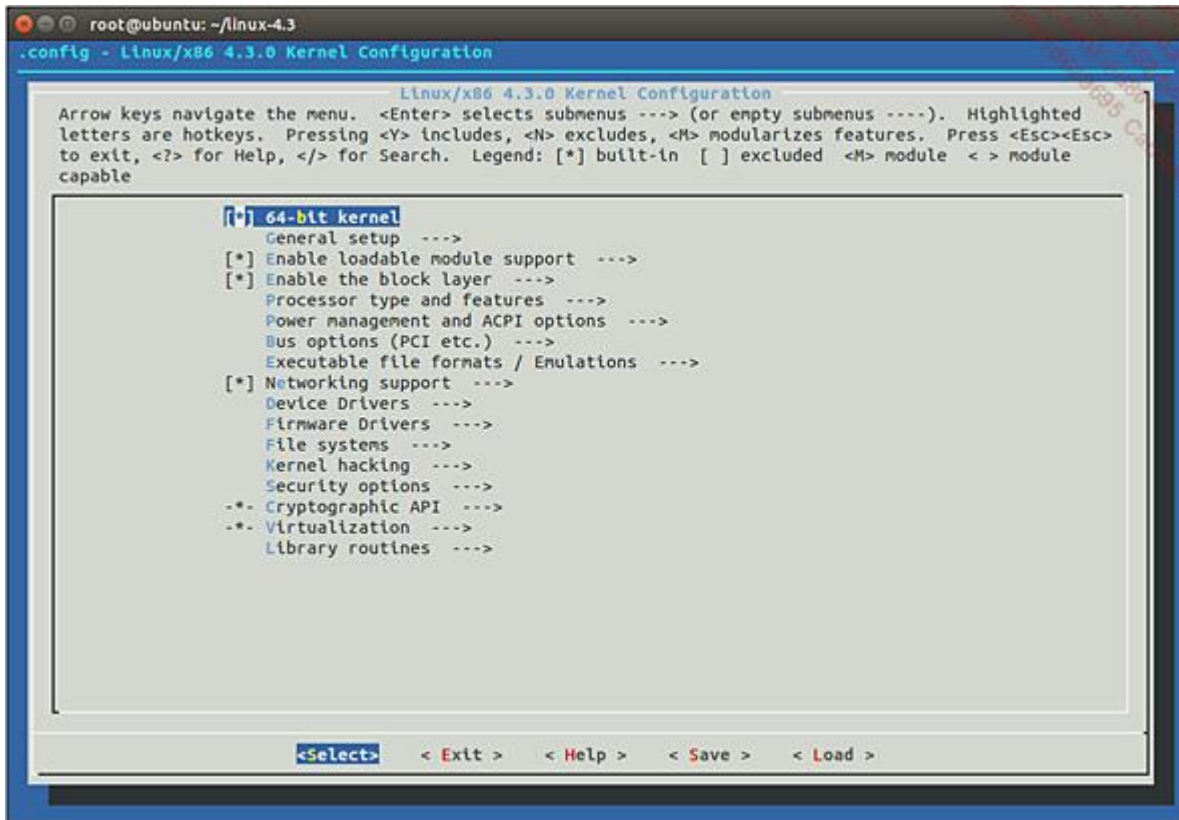
Esta herramienta tiene una ergonomía simple. Las opciones se efectúan unas tras otras:

```
# make config
scripts/kconfig/conf --oldaskconfig Kconfig
#
# using defaults found in /boot/config-3.16.0-4-amd64
#
/boot/config-3.16.0-4-amd64:1965:warning: symbol value 'm' invalid
for SCSI_DH
*
* Linux/x86 4.3.0 Kernel Configuration
*
64-bit kernel (64BIT) [Y/n/?] Y

*
* 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) []
*** Salida truncada ***
```

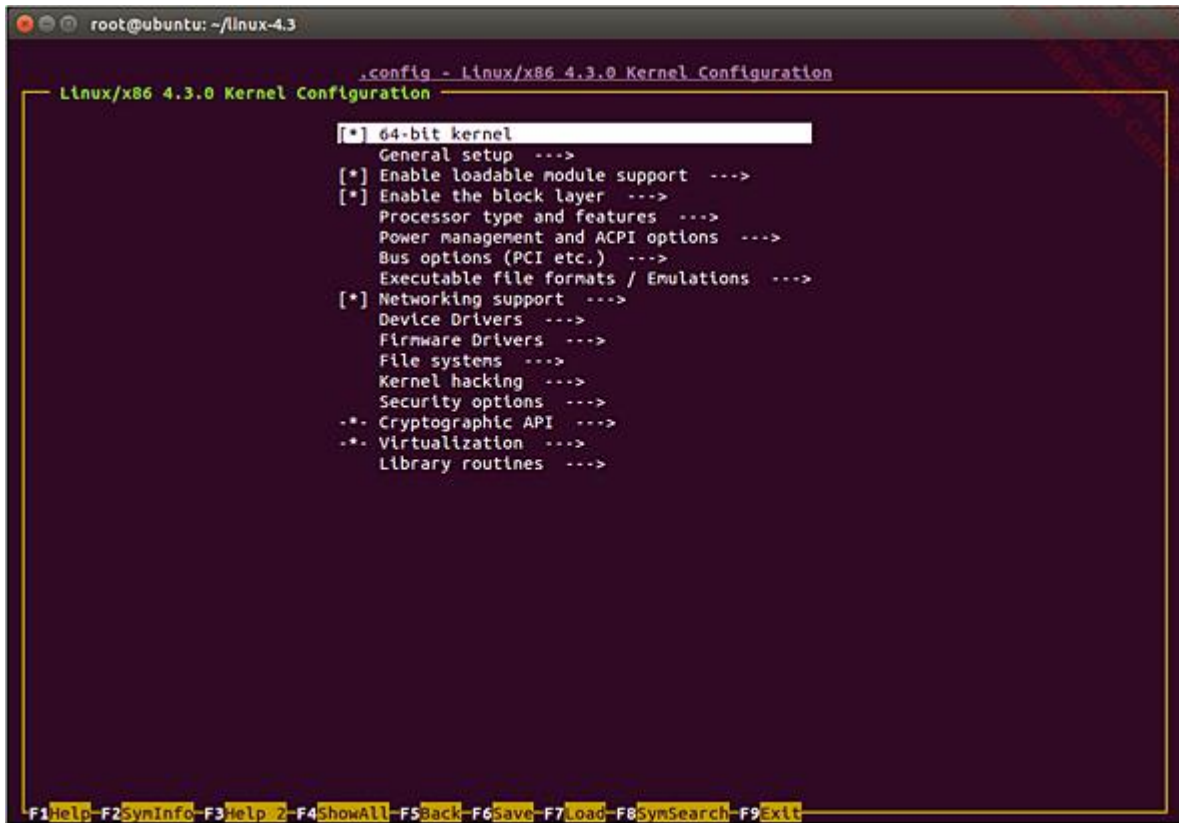
make menuconfig

Está desarrollado con la biblioteca ncurses, lo que implica la instalación de libncurses5-dev para Debian y Ubuntu o ncurses-devel.x86_64 para CentOS. Empleando una gestión por menús, ofrece una funcionalidad más amigable que make config para elegir las opciones:



make nconfig

Desarrollado con la biblioteca ncurses, make nconfig es una versión mejorada de make menuconfig:

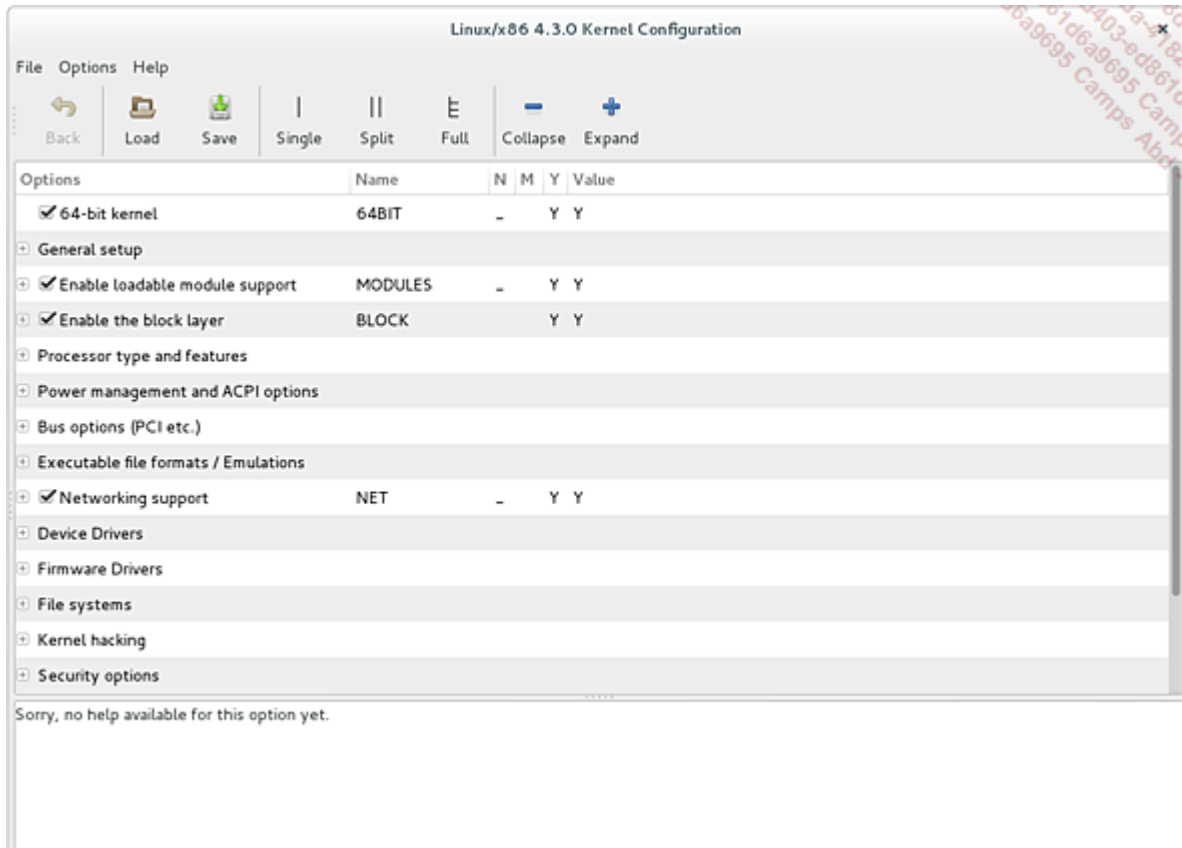


make gconfig

Requiere la instalación de libgtk2.0-dev, libglib2.0-dev y libglade2-dev para Debian.

Ubuntu Server no cuenta con un entorno gráfico nativo. Si bien es cierto que podemos añadirlo por nuestra cuenta.

Si hemos instalado GNOME 3 en nuestro sistema CentOS 7, agregue las librerías necesarias: gtk2-devel-2.24.22-5.el7_0.1.x86_64, glib2-devel-2.40.0-4.el7.x86_64 y libglade2-devel-2.6.4-11.el7.x86_64.

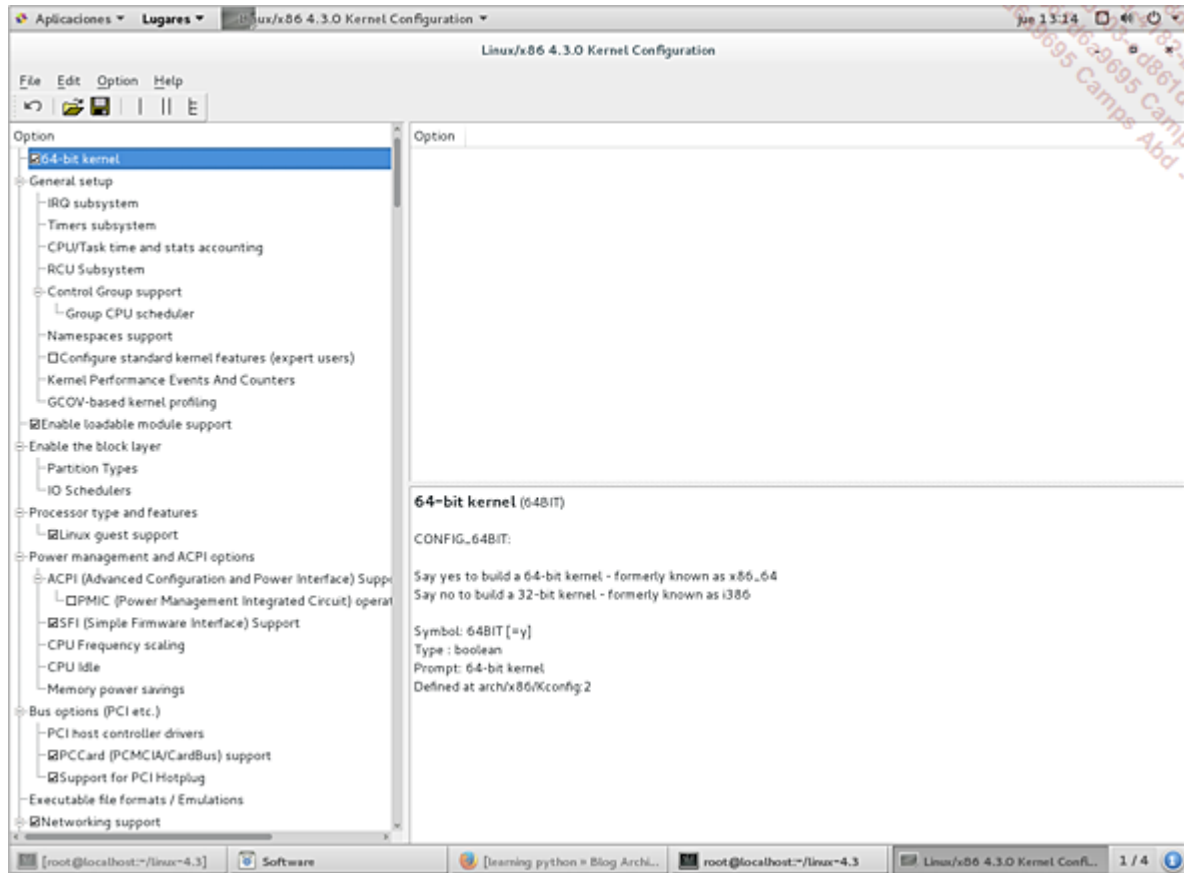


make xconfig

Desarrollado con la biblioteca Qt, requiere la instalación del paquete libqt4-dev en Debian.

Ubuntu Server no cuenta con un entorno gráfico por defecto.

La librería necesaria para CentOS 7 es qt-devel-4.8.5-8.el7.x86_64.



Si administramos el servidor en modo texto desde un terminal remoto, make menuconfig o make nconfig son probablemente las mejores opciones. Las opciones del kernel están activadas (Y), desactivadas (N) o pueden cargarse como módulos (M) y se enumeran de la siguiente manera:

- **64-bit kernel** (seleccionado por defecto) contiene el código de 64 bits del kernel de Linux. Se encuentra activado obligatoriamente.
- **General setup** contiene las opciones generales de configuración del kernel como el mecanismo de paginación de memoria virtual, la gestión de los IPC compatibles con System V, etc.
- **Enable loadable module support** (seleccionado por defecto) contiene el soporte de Loadable Kernel Modules (LKM). Este tema se aborda en el capítulo dedicado a los módulos.
- **Enable the block layer** (seleccionado por defecto) contiene el soporte para los dispositivos de tipo bloque.
- **Processor type and features** contiene el soporte para varios procesadores.

- **Power management and ACPI options** contiene la gestión de energía.
- **Bus opciones** incluye la gestión de los buses como PCI, PCMCIA, etc.
- **Executable file formats / Emulations** gestiona el uso de archivos binarios en formato ELF. El hecho de que este formato de archivo se haya convertido en un estándar hace recomendable activar esta opción.
- **Networking support** (seleccionado por defecto) contiene el soporte de las funcionalidades de red. Es una opción casi obligatoria. Es raro encontrar un servidor GNU/Linux sin servicios de red.
- **Device Drivers** gestiona las opciones relativas a la administración de dispositivos.
- **Firmware drivers** gestiona las opciones relativas a los firmwares (BIOS, EFI, DMI, etc.).
- **File systems** activa o desactiva el soporte de los sistemas de archivos (ext2, ext3, ext4, JFS, XFS, BRFS, etc.).
- **Kernel hacking** permite el seguimiento del kernel de Linux. Esta opción no se recomienda para un uso cotidiano.
- **Security options** permite el almacenamiento de claves privadas en el kernel en lugar de en los procesos.
- **Cryptographic API** (seleccionado por defecto) permite la integración del soporte de los algoritmos de encriptación SHA1, AES, MD5, MD4, etc.
- **Virtualization** (seleccionado por defecto) permite el soporte de máquinas virtuales basadas en el Kernel (KVM).
- **Library routines** ofrece la posibilidad de incluir diferentes funciones. Los componentes del kernel sin embargo no las utilizarán. Las funciones se exportan para poder ser utilizadas desde módulos de terceros.

d. Archivos de configuración

En el directorio /boot de su distribución, un archivo de texto contiene la configuración del kernel que utiliza su sistema operativo. Su nomenclatura es: Config-<versión_del_kernel><arquitectura>.

Distribución	Nombre del archivo
Debian 8	config-3.16.0-4-amd64
Ubuntu Server 14.04 LTS	config-3.13.0-48-generic
CentOS 7	config-3.10.0-229.20.1.el7.x86_64

Aquí tenemos, por ejemplo, el contenido del archivo `/boot/config-3.16.0-4-amd64` de Debian 8:

```
# cat config-3.16.0-4-amd64
#
# Automatically generated file; DO NOT EDIT.
# Linux/x86 3.16.7-ckt11 Kernel Configuration
#
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/x86_64_defconfig"
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_HAVE_LATENCYTOP_SUPPORT=y
CONFIG_MMU=y
CONFIG_NEED_DMA_MAP_STATE=y
CONFIG_NEED_SG_DMA_LENGTH=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_GENERIC_BUG=y
CONFIG_GENERIC_BUG_RELATIVE_POINTERS=y
*** Salida truncada ***
```

Antes de construir un kernel, probablemente deseemos recuperar las opciones del kernel en ejecución, por lo tanto, desde uno de los archivos mencionados anteriormente.

Con el comando `cp` basta con copiar el archivo de configuración del kernel que se encuentra situado en la carpeta `/boot` a la raíz de la carpeta de las fuentes del kernel nuevo. Hay que darle de forma obligatoria el nombre `.config`:

Ejemplo con Debian

Copiamos el archivo `/boot/config-3.16.0-4-amd64` del kernel que se está ejecutando en la carpeta `~/linux-3.10.2` y cambiamos su nombre a `.config`:

```
# cp /boot/config-3.16.0-4-amd64 ../.config
```

Otra solución es usar el comando `make defconfig`:

```
# make defconfig
*** Default configuration is based on 'x86_64_defconfig'
#
# configuration written to .config
#
```

e. Eliminar los archivos de una compilación anterior

Para limpiar las trazas de una compilación previa que podrían interferir con la nueva compilación, el comando `make` tiene dos objetivos:

- `mrproper`, que elimina todos los archivos generados, el archivo `.config` y también los archivos de respaldo para recuperar la configuración de base tal y como estaba en el archivo. Escriba `make mrproper` en un terminal:

```
# make mrproper
CLEAN    .
CLEAN    arch/x86/entry/vdso
CLEAN    arch/x86/kernel/cpu
CLEAN    arch/x86/kernel
CLEAN    arch/x86/realmode/rm
CLEAN    arch/x86/lib
CLEAN    drivers/tty/vt
CLEAN    drivers/video/logo
CLEAN    firmware
CLEAN    lib
CLEAN    security/selinux
CLEAN    usr
CLEAN    arch/x86/boot/compressed
CLEAN    arch/x86/boot
CLEAN    arch/x86/tools
CLEAN    .tmp_versions
CLEAN    scripts/basic
CLEAN    scripts/kconfig
CLEAN    scripts/mod
CLEAN    scripts/selinux/genheaders
CLEAN    scripts/selinux/mdp
CLEAN    scripts
CLEAN    include/config include/generated arch/x86/include/generated
CLEAN    .config .version Module.symvers
```

- `clean`, que elimina los archivos generados. Se conserva el archivo `.config`. Teclee `make clean` en un terminal:

```
# make clean
```

```
CLEAN .
```

```
CLEAN arch/x86/entry/vdso
```

```
CLEAN arch/x86/kernel/cpu
```

```
CLEAN arch/x86/kernel
```

```
CLEAN arch/x86/realmode/rm
```

```
CLEAN arch/x86/lib
```

```
CLEAN drivers/tty/vt
```

```
CLEAN drivers/video/logo
```

```
CLEAN firmware
```

```
CLEAN lib
```

```
CLEAN security/selinux
```

```
CLEAN usr
```

```
CLEAN arch/x86/boot/compressed
```

```
CLEAN arch/x86/boot
```

```
CLEAN arch/x86/tools
```

```
CLEAN .tmp_versions
```

f. Construir el kernel

La compilación se efectúa con el comando make. Para acelerar la compilación, podemos usar la opción -j indicando el número de CPUs más 1. Por ejemplo, para ocho cores: make -j9.

```
# make -j9
HOSTCC  scripts/basic/fixdep
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
CHK      include/generated/utsrelease.h
HOSTCC  arch/x86/tools/relocs_64.o
HOSTCC  arch/x86/tools/relocs_common.o
HOSTCC  arch/x86/tools/relocs_32.o
HOSTCC  scripts/kallsyms
HOSTCC  scripts/pnmtologo
HOSTCC  scripts/conmakehash
CC       scripts/mod/empty.o
HOSTCC  scripts/selinux/genheaders/genheaders
HOSTCC  scripts/mod/mk_elfconfig
CC       scripts/mod/devicetable-offsets.s
HOSTCC  scripts/selinux/mdp/mdp
GEN      scripts/mod/devicetable-offsets.h
HOSTCC  scripts/sortextable
MKELF    scripts/mod/elfconfig.h
HOSTCC  scripts/mod/modpost.o
HOSTCC  scripts/mod/file2alias.o
HOSTCC  scripts/mod/sumversion.o
HOSTLD   arch/x86/tools/relocs
CHK      include/generated/timeconst.h
CC       kernel/bounds.s
CHK      include/generated/bounds.h

CC       arch/x86/kernel/asm-offsets.s
*** Salida truncada ***
```

La duración de la compilación varía entre 45 minutos y 1 hora.

g. Instalar un módulo del kernel

La instalación de los módulos del kernel (Loadable Kernel Modules) se efectúa con `make modules_install`, mientras que la instalación del kernel propiamente dicho se hace con `make install`. Podemos combinar ambos dos comandos en uno solo:

```
# make modules_install install
      INSTALL crypto/echainiv.ko
      INSTALL drivers/thermal/x86_pkg_temp_thermal.ko
      INSTALL fs/efivarfs/efivarfs.ko
      INSTALL net/ipv4/netfilter/ipt_MASQUERADE.ko
      INSTALL net/ipv4/netfilter/iptable_nat.ko
      INSTALL net/ipv4/netfilter/nf_log_arp.ko
      INSTALL net/ipv4/netfilter/nf_log_ipv4.ko
      INSTALL net/ipv4/netfilter/nf_nat_ipv4.ko
      INSTALL net/ipv4/netfilter/nf_nat_masquerade_ipv4.ko
      INSTALL net/ipv6/netfilter/nf_log_ipv6.ko
      INSTALL net/netfilter/nf_log_common.ko
      INSTALL net/netfilter/nf_nat.ko
      INSTALL net/netfilter/nf_nat_ftp.ko
      INSTALL net/netfilter/nf_nat_irc.ko
      INSTALL net/netfilter/nf_nat_sip.ko
      INSTALL net/netfilter/xt_LOG.ko
      INSTALL net/netfilter/xt_addrtype.ko
      INSTALL net/netfilter/xt_mark.ko
      INSTALL net/netfilter/xt_nat.ko
      DEPMOD 4.3.0
sh ./arch/x86/boot/install.sh 4.3.0 arch/x86/boot/bzImage \
System.map "/boot"

run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.3.0
/boot/vmlinuz-4.3.0
```

```
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.3.0
/boot/vmlinuz-4.3.0
update-initramfs: Generating /boot/initrd.img-4.3.0
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.3.0
/boot/vmlinuz-4.3.0
Generating grub configuration file ...
Encontrada imagen de Linux: /boot/vmlinuz-4.3.0
Encontrada imagen de memoria inicial: /boot/initrd.img-4.3.0
Encontrada imagen de linux: /boot/vmlinuz-3.16.0-4-amd64
Encontrada imagen de memoria inicial: /boot/initrd.img-3.16.0-4-amd64
hecho
```

La carpeta /boot contiene los siguientes archivos:

- config-4.3.0
- initrd.img-4.3.0
- System.map-4.3.0
- vmlinuz-4.3.0

Se actualiza el archivo de configuración de GRUB. Bastará con reiniciar el sistema para utilizar el nuevo kernel Linux instalado.

Debian, Ubuntu y CentOS tienen un GRUB versión 2.

Ejemplo con Debian

Con el comando grep y el patrón 4.3.0, comprobamos si el archivo /boot/grub.cfg de Debian contiene la configuración de arranque del nuevo kernel.

```
# grep 4.3.0 /boot/grub/grub.cfg
echo 'Cargando Linux 4.3.0...'
linux /boot/vmlinuz-4.3.0 root=/dev/mapper/SystemVG-rootLV
ro quiet
initrd /boot/initrd.img-4.3.0
menuentry 'Debian GNU/Linux, con Linux 4.3.0' --class debian --
class gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-4.3.0-advanced-7317b5e2-ebf8-4cfb-bac2-003d23bc7d2f' {
echo 'Cargando Linux 4.3.0...'
```



```
linux    /boot/vmlinuz-4.3.0 root=/dev/mapper/SystemVG-rootLV
ro quiet

initrd   /boot/initrd.img-4.3.0

menuentry 'Debian GNU/Linux, with Linux 4.3.0 (recovery mode)'
--class debian --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-4.3.0-recovery-7317b5e2-ebf8-4cfb-bac2-
003d23bc7d2f' {

    echo    'Cargando Linux 4.3.0...'

    linux    /vmlinuz-4.3.0 root=/dev/mapper/sysVG-rootLV ro single
    initrd   /initrd.img-4.3.0
```

2. Compilar un kernel al estilo Debian

Este método, como indica su nombre, está destinado a la distribución Debian.

Efectuamos la descarga del código fuente del kernel y la extracción del archivo en el directorio del usuario /home/bob. Utilizamos las herramientas de configuración de la misma manera que durante la compilación clásica.

a. Paquetes necesarios

Los paquetes necesarios para la compilación a instalar son:

- build-essential
- kernel-package
- debconf-utils
- dpkg-dev
- debhelper
- ncurses-dev
- fakeroot

Se pueden añadir otros paquetes a la lista anterior en función de las necesidades o de las dependencias.

La ubicación para construir el kernel debe disponer de espacio en disco suficiente.

b. Privilegios de usuario

No es necesario ser el súper usuario (root) para compilar. Basta con añadir bob al grupo src con gpasswd:

Sintaxis

```
gpasswd <opción> <usuario> <grupo(s)>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-a	--add	Añadir el usuario referenciado en este grupo.
-d	--delete	Eliminar el usuario referenciado de este grupo.

Teclee en un terminal:

```
$ sudo gpasswd -a bob src
[sudo] password for bob
Añadiendo al usuario bob al grupo src
```

Para verificar:

```
$ grep src /etc/group
src:x:40:bob
```

Vamos a garantizar con el comando `chown` que el usuario `bob` y el grupo `src` son propietarios de todo el árbol de directorios.

Sintaxis

```
chown <opción> <herramienta_propietario>:<grupo_dueño> <directorio>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-R	--recursive	Modificar recursivamente archivos y directorios.

Teclee en un terminal:

```
$ sudo chown -R bob:src ./linux-3.10.2
[sudo] password for bob:
```

c. Eliminar los archivos de una compilación anterior

El comando `make-kpkg` permite realizar una limpieza antes de cualquier nueva compilación de las fuentes del kernel de Linux.

Sintaxis

```
make-kpkg <objetivo>
```

Objetivo necesario:

Objetivo	Descripción
clean	Borra todos los archivos creados en el directorio de las fuentes del kernel.

Escriba en un terminal `make-kpkg clean`:

```
# make-kpkg clean
exec make kpkg_version=13.014+nmul -f /usr/share/kernel-
package/ruleset/minimal.mk clean
===== making target minimal_clean [new prereqs: ]=====
This is kernel package version 13.014+nmul.
test ! -f .config || cp -pf .config config.precious
test ! -e stamp-building || rm -f stamp-building
test ! -f Makefile || \
    make ARCH=x86_64 distclean
```

```
make[1]: Entering directory '/home/bob/linux-4.3'
  CLEAN   scripts/basic
  CLEAN   scripts/kconfig
  CLEAN   include/config include/generated
make[1]: Leaving directory '/home/bob/linux-4.3'
test ! -f config.precious || mv -f config.precious .config
rm -f modules/modversions.h modules/ksyms.ver scripts/cramfs/cramfsck
scripts/cramfs/mkcramfs
```

d. Construir un kernel

No es necesario tener privilegios de root, ya que el comando fakeroot simula el entorno de root para el programa make-kpkg, el cual va a generar los paquetes del kernel a construir.

Sintaxis

```
fakeroot make-kpkg <opciones> <objetivos>
```

Opciones necesarias:

Opción	Descripción
--initrd	Establece que esa imagen necesita un initrd.
--append-to-version	El argumento se añade al valor de la variable EXTRAVERSION que se encuentra en el archivo Makefile. De esta forma podemos poner la fecha del día de la compilación.

Objetivos necesarios:

Objetivo	Descripción
kernel-image	Genera un paquete Debian con un núcleo (kernel) Linux, y todos los módulos definidos en el archivo de configuración del kernel .config: linux-image-3.10.2-20130726_3.10.2-20130726-10.00.Custom_amd64.deb
kernel-headers	Crea el paquete Debian de los archivos de encabezados contenidos en el kernel Linux:

	linux-headers-3.10.2-20130726_3.10.2-20130726-10.00.Custom_amd64.deb

Teclee en un terminal:

```
$ fakeroot make-kpkg --initrd --append-to-version=-$(date '+%Y%m%d') kernel-image kernel-headers
exec make kpkg_version=13.014+nmu1 -f /usr/share/kernel-package/ruleset/minimal.mk debian APPEND_TO_VERSION=-20151125 INITRD=YES
===== making target debian/stamp/conf/minimal_debian [new prereqs: ]=====
This is kernel package version 13.014+nmu1.
test -d debian          || mkdir debian
test ! -e stamp-building || rm -f stamp-building
install -p -m 755 /usr/share/kernel-package/rules debian/rules

*** Salida truncada ***

dpkg-deb: construcción del paquete « linux-headers-4.3.0-20151125 » en
« ../linux-headers-4.3.0-20151125_4.3.0-20151125-10.00.Custom_amd64.deb ».
cp -pf debian/control.dist      debian/control
make[2]: Leaving directory '/home/bob/linux-4.3'
make[1]: Leaving directory '/home/bob/linux-4.3'
```

Se crean los archivos de paquetes Debian.

```
$ find /home/bob -name *.deb
/home/bob/linux-image-4.3.0-20151125_4.3.0-20151125-10.00.Custom_amd64.deb
/home/bob/linux-headers-4.3.0-20151125_4.3.0-20151125-10.00.Custom_amd64.de
```

e. Instalar un kernel

Para instalar el kernel que se acabamos de compilar, es necesario tener privilegios de súper usuario root usando el comando siguiente:

<https://dogramcode.com/libros-sistemas>

```
$ sudo -s
```

```
[sudo] password for bob:
```

Introduzca la contraseña del usuario. Una vez hecho esto, la instalación de paquetes del kernel se realiza como sigue:

```
# dpkg -i linux*.deb
```

```
Seleccionando el paquete linux-headers-4.3.0-20160429 previamente  
no seleccionado.
```

```
(Leyendo la base de datos ... 233931 ficheros o directorios instalados  
actualmente.)
```

```
Preparando para desempaquetar linux-headers-4.3.0-20160429_4.3.0-20160429-  
10.00.Custom_amd64.deb ..
```

```
Desempaquetando linux-headers-4.3.0-20160429 (4.3.0-20160429-10.00.Custom)  
...
```

```
Seleccionando el paquete linux-image-4.3.0-20160429 previamente no  
seleccionado.
```

```
Preparando para desempaquetar linux-image-4.3.0-20160429_4.3.0-20160429-  
10.00.Custom_amd64.deb ...
```

```
Done.
```

```
Desempaquetando linux-image-4.3.0-20160429 (4.3.0-20160429-10.00.  
Custom) ...
```

```
Configurando linux-headers-4.3.0-20160429 (4.3.0-20160429-  
10.00.Custom) ...
```

```
Examining /etc/kernel/header_postinst.d.
```

```
Configurando linux-image-4.3.0-20160429 (4.3.0-20160429-10.00.  
Custom) ...
```

```
Running depmod.
```

```
Examining /etc/kernel/postinst.d.
```

```
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.3.0-  
20160429 /boot/vmlinuz-4.3.0-20160429
```

```
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.3.0-  
20160429 /boot/vmlinuz-4.3.0-20160429
```

```
update-initramfs: Generating /boot/initrd.img-4.3.0-20160429
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.3.0-
20160429 /boot/vmlinuz-4.3.0-20160429
Generating grub configuration file ...
Encontrada imagen de Linux: /boot/vmlinuz-4.3.0-20160429
Encontrada imagen de memoria inicial: /boot/initrd.img-4.3.0-20160429
Encontrada imagen de linux: /boot/vmlinuz-4.3.0
Encontrada imagen de memoria inicial: /boot/initrd.img-4.3.0
Encontrada imagen de linux: /boot/vmlinuz-3.16.0-4-amd64
Encontrada imagen de memoria inicial: /boot/initrd.img-3.16.0-4-amd64
hecho
```

El archivo /boot/grub/grub.cfg se actualiza:

```
# grep 4.3.0 /boot/grub/grub.cfg
echo      'Cargando Linux 4.3.0...'
          linux      /boot/vmlinuz-4.3.0 root=/dev/mapper/SystemVG-rootLV
```

```

ro quiet
    initrd      /boot/initrd.img-4.3.0
    menuentry 'Debian GNU/Linux, con Linux 4.3.0' --class
debian --class gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-4.3.0-advanced-7317b5e2-ebf8-4cfb-bac2-003d23bc7d2f' {
    echo      'Cargando Linux 4.3.0...'
    linux     /boot/vmlinuz-4.3.0 root=/dev/mapper/SystemVG-
rootLV ro quiet
    initrd    /boot/initrd.img-4.3.0

    menuentry 'Debian GNU/Linux, with Linux 4.3.0 (recovery
mode)' --class debian --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-4.3.0-recovery-7317b5e2-ebf8-4cfb-bac2-
003d23bc7d2f' {
    echo      'Cargando Linux 4.3.0...'
    linux     /vmlinuz-4.3.0-20151125 root=/dev/mapper/sysVG-
rootLV ro single
    initrd    /initrd.img-4.3.0-20151125

```

No queda más que reiniciar el equipo para probar el nuevo kernel instalado. El último kernel instalado se define como kernel por defecto. Podemos abrir una sesión en línea de comandos y escribir:

```

# uname -r
4.3.0

```

Desinstalar un kernel

Un kernel se instala como cualquier otro programa. No sustituye al kernel anterior pero se añade en el sistema. El menú de GRUB permite elegir durante el arranque del equipo qué núcleo utilizar.

Se aconseja tener, además del kernel utilizado en producción, un segundo kernel considerado estable y fiable en el caso de que el primero no pueda arrancar más.

Los otros kernels no necesarios se pueden desinstalar.

1. Debian y Ubuntu

a. Mostrar los kernels instalados

En la carpeta `/lib/modules`, existe una subcarpeta por cada kernel instalado. Contiene los módulos del kernel (Loadable Kernel Modules). El comando `ls` permite mostrarlos así:

```
# ls -l /lib/modules
drwxr-xr-x 4 root root 4096 abr 19 19:50 3.16.0-4-amd64
drwxr-xr-x 3 root root 4096 abr 28 20:35 4.3.0
drwxr-xr-x 3 root root 4096 abr 29 11:31 4.3.0-20160429
```

El comando `dpkg` con la opción `-l` permite mostrar los paquetes instalados. `grep` filtra los paquetes que cumplen el patrón `linux-image*` y `awk` formatea la visualización del resultado.

```
#dpkg -l |grep -i "linux-image*" | awk '{print $2}'
linux-image-3.16.0-4-amd64
linux-image-4.3.0-20160429
linux-image-amd64
```

b. Desinstalar

La desinstalación de un kernel se realiza como un programa ya que es un paquete Debian. La operación puede realizarse con `apt-get purge`:

```
# apt-get purge "linux-image-4.3.0"
```

```
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Note, selecting 'linux-image-4.3.0-20160429-dbg' for regex
'linux-image-4.3.0'
Note, selecting 'linux-image-4.3.0-20160429' for regex
'linux-image-4.3.0'
El paquete «linux-image-4.3.0-20160429-dbg» no está instalado,
no se eliminará
Los paquetes indicados a continuación se instalaron de forma
automática y ya no son necesarios.
    libasn1-8-heimdal libgssapi3-heimdal libhcrypto4-heimdal
    libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal
    libkrb5-26-heimdal libroken18-heimdal libwind0-heimdal
Utilice «apt-get autoremove» para eliminarlos.
Los siguientes paquetes se ELIMINARÁN:
    linux-image-4.3.0-20160429*
0 actualizados, 0 nuevos se instalarán, 1 para eliminar y 0 no
actualizados.
Se liberarán 10,0 MB después de esta operación.
¿Desea continuar? [S/n] s
(Leyendo la base de datos ... 248232 ficheros o directorios
instalados actualmente.)
Desinstalando linux-image-4.3.0-20160429 (4.3.0-20160429-
10.00.Custom) ...
Examining /etc/kernel/prerm.d.
Examining /etc/kernel/postrm.d .

run-parts: executing /etc/kernel/postrm.d/initramfs-
tools 4.3.0-20160429 /boot/vmlinuz-4.3.0-20160429
update-initramfs: Deleting /boot/initrd.img-4.3.0-20160429
run-parts: executing /etc/kernel/postrm.d/zz-update-grub 4.3.0-20160429
```

```
/boot/vmlinuz-4.3.0-20160429
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Encontrada imagen de linux: /boot/vmlinuz-4.3.0
Encontrada imagen de memoria inicial: /boot/initrd.img-4.3.0
Encontrada imagen de linux: /boot/vmlinuz-3.16.0-4-amd64
Encontrada imagen de memoria inicial: /boot/initrd.img-3.16.0-4-amd64
hecho
Purgando ficheros de configuración de linux-image-4.3.0-20160429
(4.3.0-20160429-10.00.Custom) ...
Examining /etc/kernel/postrm.d .
run-parts: executing /etc/kernel/postrm.d/initramfs-
tools 4.3.0-20160429 /boot/vmlinuz-4.3.0-20160429
run-parts: executing /etc/kernel/postrm.d/zz-update-grub 4.3.0-20160429
/boot/vmlinuz-4.3.0-20160429
```

2. CentOS

a. Mostrar los kernels instalados

Al igual que en Debian y Ubuntu, la carpeta `/lib/modules` tiene una carpeta por cada kernel instalado. Por lo tanto, podemos usar el comando `ls`:

```
# ls -1 /lib/modules
3.10.0-123.el7.x86_64
3.10.0-229.14.1.el7.x86_64
3.10.0-229.20.1.el7.x86_64
4.3.0
```

El comando `rpm` (Red Hat Package Manager) utilizado con la opción `-q` (`--query`) permite mostrar los paquetes kernel instalados:

```
# rpm -q kernel
kernel-3.10.0-123.el7.x86_64
```

```
kernel-3.10.0-229.14.1.el7.x86_64
```

```
kernel-3.10.0-229.20.1.el7.x86_64
```

b. Desinstalar

El comando yum (Yellowdog Updater Modified) es una capa de rpm que administra la instalación con las dependencias o la eliminación de paquetes.

Sintaxis

```
yum <acción> <paquete(s)>
```

Acción necesaria:

Acción	Descripción
remove	Elimina el paquete especificado.

Teclee en un terminal:

```
yum -y remove kernel-3.10.0-327.el7.x86_64
```

```
Complementos cargados:fastestmirror, langpacks
```

```
Resolviendo dependencias
```

```
--> Ejecutando prueba de transacción
---> Paquete kernel.x86_64 0:3.10.0-327.el7 debe ser eliminado
--> Resolución de dependencias finalizada
```

Dependencias resueltas

```
=====
Package      Arquitectura  Versión      Repositorio  Tamaño
=====
Eliminando:
kernel       x86_64        3.10.0-327.el7 @anaconda    136 M
```

Resumen de la transacción

```
=====
Eliminar 1 Paquete
```

Tamaño instalado: 136 M

*** Salida truncada ***

Al ser el comando base, rpm también puede eliminar un paquete:

```
# rpm -e kernel-2.6.32-358.0.1.el6.x86_64
```

El comando package-cleanup permite limpiar los antiguos kernels. Se encuentra por defecto en CentOS 7. Sin embargo, si no lo encontramos, podemos instalar el paquete yum-utils con yum.

Sintaxis

```
package-cleanup <opciones>
```

Opciones necesarias:

Opciones	Descripción
<code>--oldkernels</code>	Elimina los kernels antiguos.
<code>--count=<valor></code>	Indica el número de versiones de kernel a conservar.

Teclee en un terminal:

```
# package-cleanup --oldkernels --count=2
Complementos cargados:fastestmirror, langpacks
--> Ejecutando prueba de transacción
---> Paquete kernel-devel.x86_64 0:3.10.0-327.el7 debe ser eliminado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package      Arquitectura  Versión          Repositorio      Tamaño
=====
Eliminando:
kernel        x86_64        3.10.0-327.el7   @anaconda        136 M
kernel-devel  x86_64        3.10.0-123.el7   @anaconda        31 M

Resumen de la transacción

=====
Eliminar 2 Paquetes

Tamaño instalado: 136 M
*** Salida truncada ***
```

c. Limitar el número de paquetes

Podemos definir en el archivo `/etc/yum.conf` el número máximo de paquetes que pueden ser instalados al mismo tiempo por un programa.

El valor por defecto es 5:

```
# grep install /etc/yum.conf
installonly_limit=5
```

El comando `sed` permite modificar este valor.

Sintaxis

```
sed <opción> <s/texto_de_búsqueda/texto_de_reemplazo/'> <archivo>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-i	--in-place	Modifica el archivo in situ.

Para limitar el número de paquetes a 2, escribimos en un terminal:

```
# sed -i 's/installonly_limit=5/installonly_limit=2/'
/etc/yum.conf
```

y verificamos:

```
# grep install /etc/yum.conf
installonly_limit=2
```

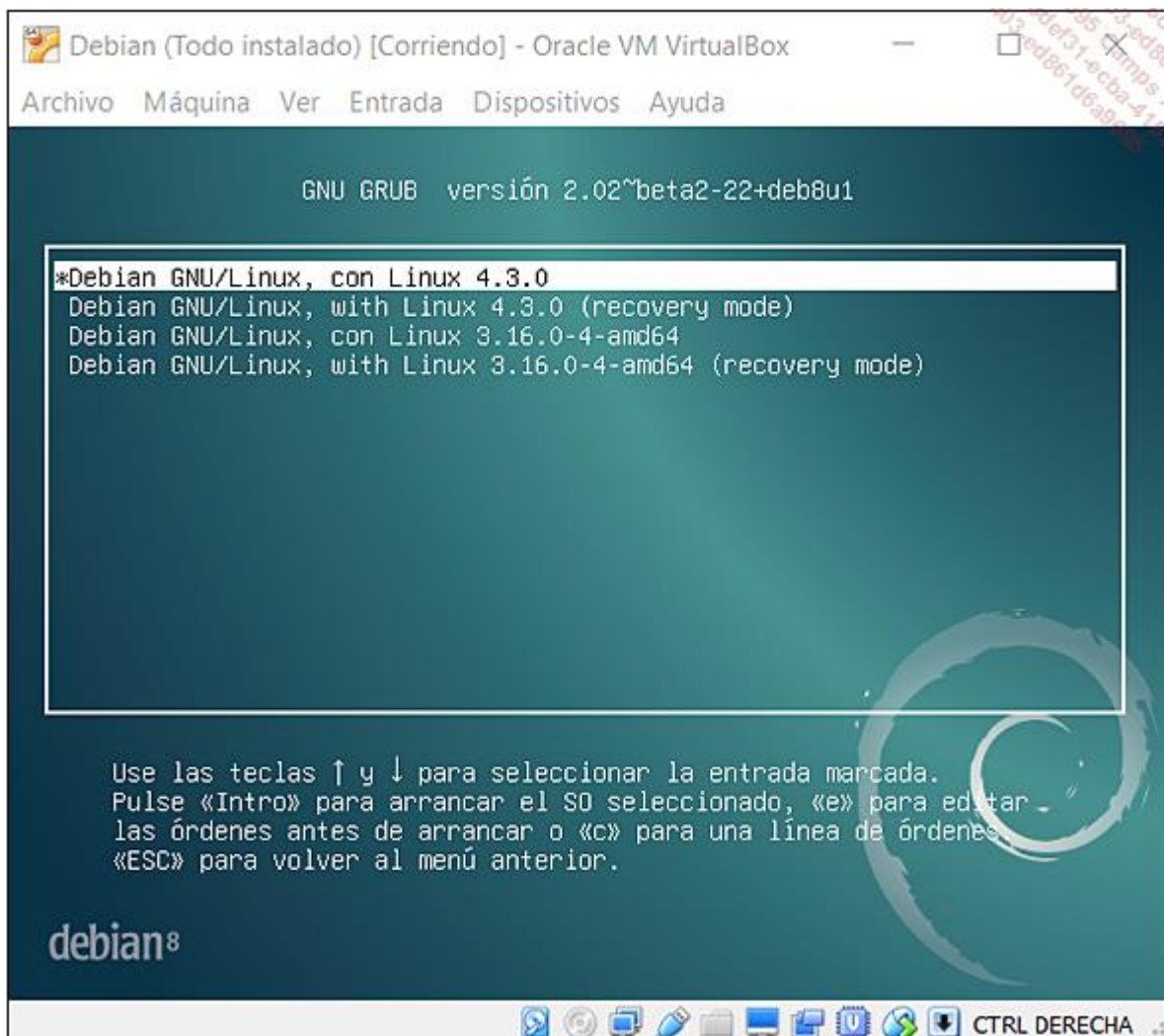
Elegir un kernel durante el arranque

Durante el arranque del equipo, pulsando una tecla del teclado se pide a GRUB que muestre en un menú la lista de kernels disponibles para cargar. Cada kernel dispone de dos modos de arranque:

- El modo normal, que es el de uso corriente;
- El modo de rescate para llevar a cabo una intervención.

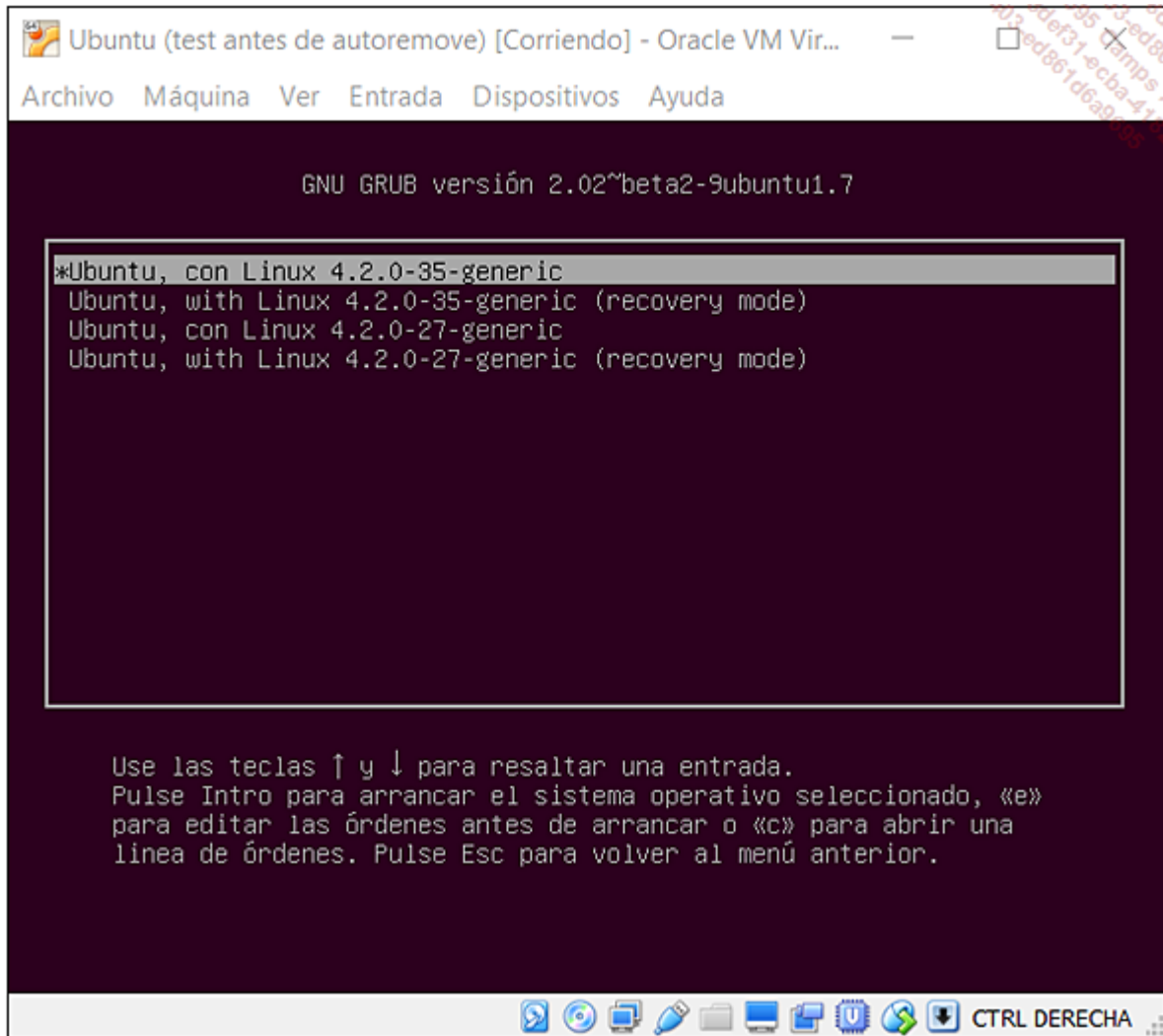
1. Debian

Al pulsar la tecla [Shift] o [Bloq Mayús] se muestra el menú de GRUB versión 2:



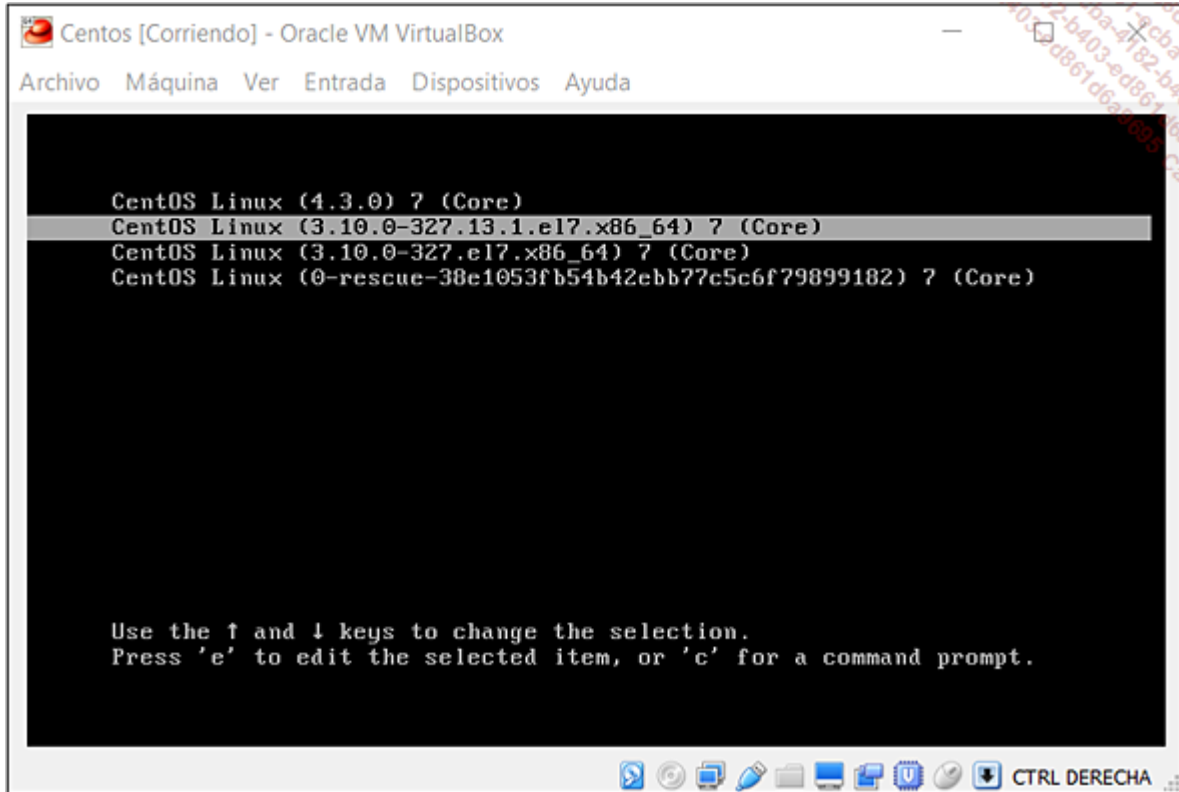
2. Ubuntu

Al ser una distribución derivada de Debian, son a su vez las teclas [Shift] o [Bloq Mayús] las que muestran el menú. La organización de este menú cambia ligeramente:



3. CentOS

Durante el arranque del equipo, un retardo de 5 segundos permite al usuario interceptar a GRUB mediante una tecla cualquiera del teclado.



4. Retardo para elegir el kernel a ejecutar

Al cabo de 5 segundos, el sistema ejecutará el kernel por defecto. Estos parámetros se definen con la variable GRUB_TIMEOUT en el archivo /etc/default/grub:

```
# cat /boot/grub/grub.conf

GRUB_TIMEOUT=5

GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"

GRUB_DEFAULT=saved

GRUB_DISABLE_SUBMENU=true

GRUB_TERMINAL_OUTPUT="console"

GRUB_CMDLINE_LINUX="vconsole.keymap=es rd.lvm.lv=sysVG/swap
vconsole.font=latarcyrheb-sun16 crashkernel=auto
rd.lvm.lv=sysVG/root
rhgb quiet net.ifnames=0"

GRUB_DISABLE_RECOVERY="true"
```

Si cambiamos el valor de GRUB_TIMEOUT, actualizamos la configuración de GRUB.

Debian y Ubuntu

```
# update-grub2 ...

Generating grub configuration file ...

Encontrada imagen de Linux: /boot/vmlinuz-3.13.0-48-generic
Encontrada imagen de memoria inicial: /boot/vmlinuz-3.13.0-48-generic
Encontrada imagen de memoria inicial: /boot/vmlinuz-3.13.0-43-generic
Encontrada imagen de memoria inicial:
/boot/initramfs-3.13.0-43.1.el7.x86_64.img
Found memtest86+ image: /memtest86+.elf
Found memtest86+ image: /memtest86+.bin
hecho
```

CentOS

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.3.0
Found initrd image: /boot/initramfs-4.3.0.img
Found linux image: /boot/vmlinuz-3.10.0-229.20.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-
229.20.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-229.14.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-
229.14.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-
85d58cfe9e95420caf192d072326775e
Found initrd image: /boot/initramfs-0-rescue-
85d58cfe9e95420caf192d072326775e.img
done
```

Kernel panic

El kernel panic es un error del kernel que puede ocurrir en diversas circunstancias.

Existen dos categorías de kernel panic:

- Hardware (conocido con el nombre de Aieee!): un componente de hardware es defectuoso, o no está soportado por el kernel...
- Software (conocido con el nombre de Oops!), es decir, los errores en el código del kernel, las modificaciones erróneas, una compilación o una instalación del kernel incorrectas, etc.

Un kernel panic puede aparecer en diferentes momentos y provocar una indisponibilidad del equipo. Puede aparecer posiblemente un mensaje de error en la pantalla. Los LED del teclado también pueden parpadear... Podemos considerar que el nivel de gravedad es alto.

1. Características

Si el funcionamiento normal del sistema GNU/Linux se ve perturbado, entonces:

- Podemos intentar arrancar el sistema operativo con otro kernel que se instalaría.
- El rescate se realiza desde un CD, DVD o una llave USB que contienen un sistema "live" GNU/Linux.

Debemos determinar si:

- El problema se produce en el primer arranque del kernel tras su instalación. Hay que eliminarlo y sin duda volver a compilar con las opciones correctas y luego efectuar una nueva instalación.
- El kernel no funciona de forma sistemática o de vez en cuando al arrancar el equipo.
- El kernel se bloquea de forma sistemática al ocurrir un evento específico o de forma aleatoria durante el funcionamiento del sistema operativo.

2. Frecuencia del problema

He aquí un conjunto de cuestiones a plantearse:

- ¿Desde hace cuánto tiempo tenemos este problema?
- ¿Es la primera vez? En caso contrario, ¿es un fallo recurrente? ¿Con qué frecuencia?
- ¿Podemos reproducir el problema?
- ¿Se encuentra el sistema completamente inutilizable?
- ¿Hemos cambiado algo justo antes de la aparición del problema? En caso afirmativo, ¿qué?
- ¿Tiene constancia escrita de las intervenciones en este equipo?

3. Análisis de la fuente del problema

Para determinar fácilmente la fuente del problema, asegúrese de que el kernel no está en modo «quiet» en el archivo `/etc/default/grub` para GRUB versión 2.

Debian y Ubuntu

Debemos sustituir con un editor de texto la siguiente línea:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
```

Por:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

```
...
```

CentOS

Hay que quitar la palabra `quiet` en la siguiente línea:

```
GRUB_CMDLINE_LINUX="vconsole.keymap=fr rd.lvm.lv=sysVG/swap  
vconsole.font=latarcyrheb-sun16 crashkernel=auto rd.lvm.lv=sysVG/root  
rhgb quiet"
```

Después de grabar el archivo, ejecute en un terminal `update-grub` para Debian y Ubuntu o `grub2-mkconfig -o /boot/grub2/grub.cfg` para CentOS.

Su modificación se tendrá en cuenta en el próximo arranque.

Durante el arranque, observe los mensajes del kernel anteriores al kernel panic. Anote toda la información que considere pertinente.

4. Análisis del hardware

El kernel puede estar en buen estado y caer a causa de un componente hardware defectuoso. Podemos encender el equipo y arrancar con un sistema «Live» GNU/Linux.

Podemos ejecutar software del tipo `memtest86+` para realizar un diagnóstico completo de la memoria.



Para más información, consulte el capítulo Solución de problemas hardware.

La lectura de los registros del sistema situados en `/var/log` también nos puede proporcionar pistas:

Distribución	Nombre del registro	Descripción
Debian	syslog	Registro estándar del sistema.
	mensajes	Información general del sistema.
	kern.log	Información del kernel de Linux.
	dmesg	Muestra el buffer de mensajes del kernel.
Ubuntu Server	syslog	Registro estándar del sistema.
	kern.log	Información del kernel de Linux.
	dmesg	Muestra el buffer de mensajes del kernel.
CentOS	mensajes	Información general del sistema.
	dmesg	Muestra el buffer de mensajes del kernel.

Con respecto a CentOS, le falta el registro `kern.log`. Podemos cambiar con un editor de texto el archivo de configuración del servicio de registro de sucesos `/etc/rsyslog.conf` para crear el registro `/var/log/kern.log`.

Basta con modificar la línea:

```
#kern.*                                /dev/console
```

Y añadir a continuación:

```
kern.*                                /var/log/kern.log
```

De la siguiente manera:

```
# cat /etc/rsyslog.conf
# rsyslog v5 configuration file

# For more information see /usr/share/doc/rsyslog-
*/rsyslog_conf.html
# If you experience problems, see
http://www.rsyslog.com/doc/troubleshoot.html
...
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                          /dev/console
kern.*                          /var/log/kern.log
...
```

Después, debemos configurar el rotado del registro en la carpeta `/etc/logrotate.d/`. Para ello, creamos con un editor de texto un nuevo archivo que llamaremos `kernel` y luego escribimos estas líneas:

```
/var/log/kernel.log {
    missingok
    notifempty
    size 30k
    yearly
    create 0600 root root
}
```

Grabamos el archivo y reiniciamos el servicio `rsyslog`:

```
[root@localhost logrotate.d]# service rsyslog restart
Redirecting to /bin/systemctl restart rsyslog.service
```


5. Análisis de software

a. Gestor de arranque

Verifique que la configuración del gestor de arranque es correcta porque suele ser una causa frecuente.



Para más información, consulte el capítulo Secuencia de arranque.

b. Origen del kernel

Si el kernel proviene de una actualización de la distribución, consulte:

- el código fuente y también el registro de cambios (changelog),
- los informes de bugs,
- las listas de distribución de correo (mailing lists),
- los documentos oficiales o extraoficiales,
- los IRC (Internet Relay Chat),
- los foros...

Si hemos compilado el kernel, vuelva a repetir las diferentes etapas para detectar un posible error en la configuración del kernel. Analice el archivo `.config`.

c. Módulos

¿Tenemos un módulo del kernel (Loadable kernel modules) defectuoso? Descargamos el módulo afectado y probamos de nuevo. Lea los registros del sistema del cuadro anterior.



Para más información, consulte el capítulo Módulos.

6. Buscar soluciones

Teniendo en mente la calidad y confianza de los datos que podemos recoger a través de Internet, podemos buscar si el problema es conocido o no. ¿Encontramos Soluciones? ¿Hemos encontrado solo una solución para evitar el problema (workaround)?

Debemos analizar las soluciones y comprender el desarrollo del procedimiento. Nunca efectuar una intervención a ciegas. Pruebe a continuación las soluciones de una en una.

7. Aplicar la solución

Una vez que hemos comprendido lo que ocurre y que hemos verificado la solución, entonces la podremos aplicar. ¡Y no olvide crear documentación sobre el tema!

Ejercicio

Vamos a utilizar la distribución de nuestra elección.

1. Abra una sesión como root y póngase en línea de comandos.
2. Descargue el último kernel de Linux en el directorio /root.
3. Descomprima el archivo y entre en el directorio Linux.
4. Asegúrese de disponer de todos los requerimientos de software.
5. Elija el archivo de configuración del kernel activo.
6. Compile el kernel (método clásico) y luego instálelo.
7. Reinicie el sistema con el nuevo kernel.

Introducción

Un módulo o Loadable Kernel Module (LKM) en inglés es una parte de código que puede cargarse y descargarse en el kernel de Linux bajo demanda. Le aporta alguna funcionalidad como un controlador de dispositivo, un cortafuegos (firewall), un protocolo de red, etc.

Los módulos se ejecutan en modo kernel (Anillo 0) y no en el modo de usuario (Anillo 3).

El hecho de crear una funcionalidad en forma de módulo en lugar de integrarla en el kernel evita sobrecargarlo. No es necesario incluir en el núcleo el código que se utiliza prácticamente nunca.

Terceros también pueden desarrollar módulos que se proporcionan por separado. El administrador puede instalarlos y cargarlos sin necesidad de recompilar el kernel.

Compilar e instalar un módulo

Un módulo se compila para una versión específica del kernel de Linux.

Si durante la actualización de su sistema se instala una nueva versión del kernel, todos los módulos proporcionados por la distribución se compilarán de nuevo. Sin embargo, los módulos que haya añadido personalmente no se recompilarán. Necesitará compilarlos de forma manual.

1. Requisitos previos a la recompilación

La compilación requiere un conjunto de herramientas como el compilador gcc, y la utilidad make...

a. Debian y Ubuntu

Debemos instalar los paquetes gcc, build-essential y los archivos de cabecera (headers del) kernel:

```
# apt-get install gcc build-essential linux-headers-$(uname -r)
```

b. Red Hat

Instale las herramientas de desarrollo:

```
# yum groupinstall "Development tools"
```

2. Escribir un módulo "Hello World"

Para comprender la implementación de un LKM vamos a diseñar en lenguaje C un módulo simple y tradicional en el mundo del desarrollo: «Hello world».

El objetivo de este último es enviar un mensaje al registro del sistema `/var/log/syslog` durante su carga o descarga.

a. Escribir el código fuente

Empleando un editor de texto como vi o nano, creamos el archivo `hello.c`:

```
#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("John Doe");
MODULE_DESCRIPTION("Module hello world");
MODULE_VERSION("Version 1.00");

int init_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función init_module().\n");
    ;
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función cleanup_module().\n");
}
}
```

Las dos primeras líneas del código fuente incluyen dos archivos de cabecera:

- `module.h` para todos los módulos.
- `kernel.h` para la definición de macros como `KERN_INFO`.

Las macros `MODULE_LICENSE`, `MODULE_AUTHOR`, `MODULE_DESCRIPTION` y `MODULE_VERSION` definen la información acerca del módulo.

`int init_module(void)` y `void cleanup_module(void)` son dos funciones necesarias, ya que la primera se ejecutará durante la carga y la segunda durante la descarga del módulo.

La función `printk()` muestra los mensajes en el registro del kernel. Al estar en un nivel bajo (espacio kernel) no podemos utilizar la función `printf()` definida en el archivo `stdio.h` que muestra una cadena de caracteres por pantalla. Esta última se puede usar en el espacio de usuario.

b. Crear el archivo Makefile

El archivo Makefile contiene las instrucciones utilizadas por `make` durante la construcción de un ejecutable, de una librería o de un módulo.

Escriba las siguientes líneas en un editor de texto y guarde el archivo con el nombre `Makefile`:

```
obj-m += hello.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD)
modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

install:
    cp ./hello.ko /lib/modules/$(shell uname
-r)/kernel/drivers/misc
```

`obj-m` (object module) significa que hay un objeto llamado `hello.o` que se construirá a partir de `hello.c`.

Luego, `Makefile` contiene tres objetivos:

- `all`:
- `clean`:
- `install`:

El objetivo `all`: construye el módulo. Si solo escribe `make`, se aplicará el primer objetivo del archivo `Makefile`, en este caso `all`. Es lo mismo que escribir `make all`.

El objetivo `clean`: limpia los archivos de la construcción de una compilación previa. Por ejemplo, si una compilación efectuada con `make` termina con un error. Corregimos los errores

encontrados. Antes de volver a ejecutar el comando make, debemos limpiar los archivos de la anterior compilación llamando a make clean.

Si ejecutamos make install, el objetivo install: instala el ejecutable en los directorios especificados.

c. Compilar el código fuente hello.c

Escriba make para construir el módulo:

```
# make
make -C /lib/modules/3.16.0-4-amd64/build M=/root/lkm modules
make[1]: Entering directory '/usr/src/linux-headers-3.16.0-4-amd64'
Makefile:10: *** mixed implicit and normal rules: deprecated syntax
make[1]: Entering directory '/usr/src/linux-headers-3.16.0-4-amd64'
  CC [M]  /root/lkm/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/lkm/hello.mod.o
  LD [M]  /root/lkm/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-4-amd64'
```

La creación ha generado varios archivos, ya que esta se realiza en diferentes etapas. El archivo del módulo propiamente dicho es hello.ko:

```
# ls
hello.c  hello.mod.c  hello.o  Makefile  modules.order  hello.ko
hello.mod.o  Module.symvers
```

Cargar un módulo

La carga de un módulo se realiza con los comandos `insmod` o `modprobe`.

1. Instalar el módulo hello

Ejecute `make install`:

```
# make install
cp ./hello.ko /lib/modules/3.16.0-4-amd64/kernel/drivers/misc
```

El archivo `hello.ko` se copiará en la carpeta `/lib/modules/$(uname -r)/kernel/drivers/misc`:

```
# ls /lib/modules/$(uname -r)/kernel/drivers/misc
ad525x_dpot-i2c.ko    apds9802als.ko    c2port    enclosure.ko
ibmasm                isl29020.ko       phantom.ko  ts12550.ko
ad525x_dpot.ko        apds990x.ko       cb710      hello.ko
```

2. insmod

El comando `insmod` acepta cualquier ubicación para el módulo, el cual no tiene que estar necesariamente en el directorio `/lib/modules/$(uname -r)`. Sin embargo, las dependencias no se cargan de forma automática.

Utilizado por `modprobe`, es un programa de bajo nivel para cargar los módulos.

Sintaxis

```
insmod <nombre_del_modulo_a_cargar>
```

Cargar el módulo `hello.ko`

```
# insmod /lib/modules/$(uname -r)/kernel/drivers/misc/hello.ko
```

3. Gestionar las dependencias

El archivo `modules.dep` contiene la lista de dependencias de carga de cada módulo utilizado por `modprobe`.

Utilice el comando `depmod` cuando se produzca un cambio en los módulos para asegurarse de que todas las dependencias estén cargadas. Este comando reconstruye el archivo `modules.dep` comprobando todos los módulos disponibles.

Sintaxis

```
depmod <option> <kernel>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-a	--all	Sondea todos los módulos.

Ejemplos:

Explorar todos los módulos disponibles, garantizando que todas las dependencias se cargarán:

```
# depmod -a
```

depmod puede también generar el archivo de dependencias de otro kernel diferente al kernel actual especificando la versión:

```
# depmod -a 3.16.0-4
```

4. modprobe

El comando modprobe carga los módulos del kernel a partir del archivo `/lib/modules/$(uname-r)/modules.dep.bin`. Sin embargo, si este no existe, se emplea `/lib/modules/$(uname-r)/modules.dep`. Así se cargan las dependencias.

Por lo general, se prefiere el comando modprobe a insmod.

Sintaxis

```
modprobe <nombre_del_modulo_a_cargar>
```

Cargar el módulo `hello.ko`

```
# depmod -a
# modprobe hello
```

depmod -a reconstruye el archivo `modules.dep` comprobando todos los módulos disponibles; en caso contrario, obtendremos un mensaje de error:

```
# modprobe hello
```



```
FATAL: Module hello not found.
```

Compruebe que el módulo está cargado con lsmod:

```
# lsmod |grep hello  
hello                12388  0
```

Verifique en el registro /var/log/messages para Debian y CentOS:

```
# cat /var/log/messages  
*** salida truncada ***  
  
Nov 12 15:29:27 Debsrv kernel: [ 3077.970814] [Hello world] - La  
función init_module() ha sido llamada
```

O para Ubuntu, /var/log/syslog:

```
# cat /var/log/syslog  
*** salida truncada ***  
  
Nov 12 15:42:08 srv01 kernel: [ 4129.962167] [Hello world] - La  
función init_module() ha sido llamada
```

Si vuelve a iniciar su equipo, el módulo no se cargará.

5. Cargar un módulo durante el arranque

a. Debian y Ubuntu

Añada el nombre del módulo al archivo `/etc/modules`:

```
# /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.

Loop
hello
```

b. CentOS

El archivo `/etc/modules` no existe. Debemos crear un archivo de script con la extensión `.modules` en la carpeta `/etc/sysconfig/modules`. Debe contar con permisos de ejecución.

El contenido de este archivo incluye al menos `modprobe` o `insmod` seguido del nombre del módulo.

Creación del archivo `hello.modules`

Empleando un editor de texto como `vi` o `nano`, creamos el archivo:

```
# nano /etc/sysconfig/modules/hello.modules
```

El contenido del archivo comienza con un shebang para definir el intérprete de comandos a emplear en la ejecución del script. Luego, el comando `modprobe` carga el módulo `hello.ko`:

```
#!/bin/sh

/sbin/modprobe hello > /dev/null 2>&1
```

Guardamos el archivo y le asignamos permisos de ejecución:

```
# chmod 755 /etc/sysconfig/modules/hello.modules
```

Descargar un módulo

`rmmod` descarga un módulo del kernel si no existe ningún proceso utilizándolo. Si este fuera el caso, añade la opción `-w` (o `--wait`). De esta forma, el proceso queda aislado a la espera de que no se utilice más y luego se descargará.

Sintaxis

```
rmmod <opción> <nombre_del_módulo_a_descargar>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-f	--force	Fuerza la descarga del módulo.
-w	--wait	Aísla al módulo y vigila que el módulo no vuelva a utilizarse.

`modprobe` puede descargar un módulo de la misma forma que `rmmod`.

Sintaxis

```
modprobe <opciones> <nombre_del_módulo_a_descargar>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-r	--remove	Descarga el módulo correspondiente.
-v	--verbose	Modo largo (verbose).

Descargar el módulo `hello`

```
# rmmod hello
```

O bien:

```
# modprobe -rv hello
rmmod hello
```

Verifique con `lsmod` y `grep` que el módulo se ha descargado de forma correcta:

```
# lsmod |grep hello
```

Consulte el registro `/var/log/messages` para Debian y CentOS:

```
# cat /var/log/messages
```

```
*** salida truncada ***
```

```
Nov 12 15:58:52 Debsrv kernel: [ 4842.559549] [Hello world] - La  
función cleaup_module() ha sido llamada
```

O para Ubuntu, `/var/log/syslog`:

```
# cat /var/log/syslog
```

```
*** salida truncada ***
```

```
Nov 12 16:01:04 srv01 kernel: [ 5267.293122] [Hello world] - La  
función cleanup_module() ha sido llamada
```



No olvide modificar el archivo `/etc/modules` para evitar la carga de este módulo en el siguiente arranque del sistema operativo.

Mostrar los módulos

1. Mostrar todos los módulos disponibles

Para identificar los módulos disponibles, ya no existe la opción -l del comando modprobe.

Hay que recorrer con find la carpeta /lib/modules/\$(uname -r) y registrar los archivos cuya extensión es .ko. awk permitirá mostrar solo el nombre del archivo:

```
# find /lib/modules/$(uname -r) -name '*.ko' | awk -F "/" '{print $NF}'
cast5_generic.ko
crct10dif_common.ko
rmd160.ko
crc32c_generic.ko
vmac.ko
cast6_generic.ko
crct10dif_generic.ko
pcbc.ko
serpent_generic.ko
rmd128.ko
sha256_generic.ko
khazad.ko
md4.ko

*** salida truncada ***
```

2. Mostrar los módulos cargados

lsmod muestra los módulos cargados. Los datos proceden del archivo /proc/modules:

```
# lsmod
Module                Size  Used by
ppdev                 12763  0
lp                   13025  0
bnep                  17535  2
rfcomm                33424  0
bluetooth             170002  10 bnep,rfcomm
rfkill                19242  2 bluetooth
binfmt_misc           12925  1
uinput                17439  1
nfsd                  192007  2
auth_rpcgss           39085  1 nfsd
oid_registry          12419  1 auth_rpcgss
nfs_acl               12511  1 nfsd
nfs                   105982  0
lockd                 55577  2 nfs,nfsd
fscache               37551  1 nfs
sunrpc                160406  6 nfs,nfsd,auth_rpcgss,lockd,nfs_acl
loop                  22869  0
joydev                17232  0
sg                    26095  0

*** salida truncada ***
```

Las cuatro columnas muestran respectivamente:

- el nombre del módulo,
- el tamaño,
- el número de llamadas,
- los módulos de los que depende.

Obtener los datos

modinfo muestra los datos de un módulo como el nombre del archivo y su ubicación, su licencia, una descripción, el o los autores y las dependencias de esta versión.

Sintaxis

```
modinfo <nombre_de_módulo>
```

Los datos del módulo ext3

```
# modinfo ext3
filename:      /lib/modules/3.16.0-4-amd64/kernel/fs/ext4/ext4.ko
license:      GPL
description:   Fourth Extended Filesystem
author:       Remy Card, Stephen Tweedie, Andrew Morton,
Andreas Dilger, Theodore Ts'o and others
alias:        fs-ext4
alias:        ext3
alias:        fs-ext3
alias:        ext2
alias:        fs-ext2*
depends:       mbcache,jbd2,crc16
intree:       Y
vermagic:     3.16.0-4-amd64 SMP mod_unload modversions
```

Las dependencias también pueden mostrarse con modprobe.

Sintaxis

```
modprobe <opción> <nombre_del_módulo>
```

Opción:

Opción larga	Descripción
--show-depends	Muestra las dependencias de un módulo.

Ejemplo

```
# modprobe --show-depends ext4
insmod /lib/modules/3.16.0-4-amd64/kernel/fs/jbd2/jbd2.ko
insmod /lib/modules/3.16.0-4-amd64/kernel/fs/mbcache.ko
insmod /lib/modules/3.16.0-4-amd64/kernel/lib/crc16.ko
insmod /lib/modules/3.16.0-4-amd64/kernel/fs/ext4/ext4.ko
```

O bien con lsmod:

```
# lsmod |grep ext4
ext4                473802  4
crc16                12343  1 ext4
mbcache              17171  1 ext4
jbd2                 82413  1 ext4
```

Podremos organizar la visualización de la información de los módulos: nombre del módulo y ubicación, descripción y dependencias utilizando los comandos avanzados del shell. Por ejemplo:

- cut muestra zonas específicas de un archivo de texto. -d define el separador de campo y -f1 extrae el primer campo.
- sed (Stream Editor) permite manipular el texto.
- egrep (o grep -E) muestra las líneas que corresponden a un patrón.
- Expresiones regulares y expresiones regulares extendidas...

Primer ejemplo

```
# modinfo $(cut -d' ' -f1 /proc/modules) | sed '/^dep/s/$/\n/;
/^file\|^desc\|^dep!/d'

filename: /lib/modules/2.6.32-5-amd64/kernel/drivers/misc/hello.ko
description: Un ejemplo de LKM
depends:

filename: /lib/modules/2.6.32-5-amd64/kernel/drivers/cpufreq/
cpufreq_powersave.ko
description: CPUfreq policy governor 'powersave'
depends:
...
*** Salida truncada ***
```

Segundo ejemplo

```
# lsmod | cut -d' ' -f1 | xargs modinfo | egrep '^file|^desc|^dep'
| sed -e'/^dep/s/$/\n/g'

filename: /lib/modules/2.6.32-5-amd64/kernel/drivers/misc/hello.ko
description: Un ejemplo de LKM
depends:

filename: /lib/modules/2.6.32-5-amd64/kernel/drivers/cpufreq/
cpufreq_powersave.ko
description: CPUfreq policy governor 'powersave'
depends:
...
*** Salida truncada ***
```

Tercer ejemplo

```
# find /lib/modules/$(uname -r) -name '*.ko' | xargs modinfo
filename:      /lib/modules/3.10.2-
20130726/kernel/net/bluetooth/rfcomm/rfcomm.ko
alias:        bt-proto-3
license:      GPL
version:      1.11
description:   Bluetooth RFCOMM ver 1.11
author:       Marcel Holtmann <marcel@holtmann.org>
srcversion:   7D57E08D7FBF2D6B1E7BFA7
depends:       bluetooth
intree:       Y
vermagic:     3.10.2-20130726 SMP mod_unload modversions
parm:        disable_cfc:Disable credit based flow control
(bool)
parm:        channel_mtu:Default MTU for the RFCOMM channel
(int)
parm:        l2cap_mtu:Default MTU for the L2CAP connection
(uint)
parm:        l2cap_ertm:Use L2CAP ERTM mode for connection
(bool)
filename:      /lib/modules/3.10.2-
20130726/kernel/net/bluetooth/cmtcp/cmtcp.ko
alias:        bt-proto-5
license:      GPL
version:      1.0
description:   Bluetooth CMTP ver 1.0
author:       Marcel Holtmann <marcel@holtmann.org>
srcversion:   8369758E5A8D8842F5E5A94
depends:       kernelcapi,bluetooth
intree:       Y
vermagic:     3.10.2-20130726 SMP mod_unload modversions
*** salida truncada ***
```

Bloquear un módulo

Por problemas de estabilidad, compatibilidad con otros programas o cualquier otra causa, un módulo puede ser bloqueado o puesto en una «blacklist».

Tenemos la posibilidad de crear nuestro propio archivo para bloquear los módulos que deseamos en la carpeta `/etc/modprobe.d`. Se aconseja nombrarlo de manera explícita y este debe tener la extensión `.conf`.

La carpeta `/etc/modprobe.d` no tiene el mismo contenido por defecto de una distribución a otra.

Debian

```
# ls -l /etc/modprobe.d/
total 4
-rw-r--r-- 1 root root 390 abril 16 17:52 fbdev-blacklist.conf
```

Ubuntu Server

```
# ls -l /etc/modprobe.d/
total 44
drwxr-xr-x 2 root root 4096 sept. 11 16:14 ./
drwxr-xr-x 90 root root 4096 oct. 10 07:42 ../
-rw-r--r-- 1 root root 325 abr 10 2014 blacklist-ath_pci.conf
-rw-r--r-- 1 root root 1603 abr 10 2014 blacklist.conf
-rw-r--r-- 1 root root 210 abr 10 2014 blacklist-firewire.conf
-rw-r--r-- 1 root root 677 abr 10 2014 blacklist-framebuffer.conf
-rw-r--r-- 1 root root 583 abr 10 2014 blacklist-rare-network.conf
-rw-r--r-- 1 root root 1077 abr 10 2014 blacklist-watchdog.conf
-rw-r--r-- 1 root root 456 abr 14 2014 fbdev-blacklist.conf
-rw-r--r-- 1 root root 347 abr 10 2014 iwlwifi.conf
-rw-r--r-- 1 root root 104 abr 10 2014 mlx4.conf
```

CentOS

```
# ls -l /etc/modprobe.d/
total 0
```

La sintaxis de este último es `blacklist <nombre-del-controlador>`, como por ejemplo: `blacklist pcspkr`.

Bloquear el módulo hello

Cree un archivo blacklist-hello.conf en la carpeta /etc/modprobe.d/:

```
# Bloquear el módulo hello.ko
blacklist hello
```

Registro del sistema

Cuando el sistema operativo arranca, el kernel de Linux se carga en memoria. Cada controlador de dispositivo cargado hace un intento de detección del equipo tratado. Si la operación tiene éxito, entonces un mensaje de diagnóstico detalla lo que se ha encontrado.

Podemos consultar, empleando el comando dmesg, el registro /var/log/dmesg que contiene los mensajes del buffer del kernel.

Búsqueda de información sobre una tarjeta de red Intel

```
# dmesg |grep e1000
[ 4.665632] e1000: Intel(R) PRO/1000 Network Driver - version 7.3.21-
k8-NAPI
[ 4.665654] e1000: Copyright (c) 1999-2006 Intel Corporation.
[ 5.052298] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit)
08:00:27:3a:17:d8
[ 5.052304] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network
Connection
[ 5.441683] e1000 0000:00:08.0 eth1: (PCI:33MHz:32-bit)
08:00:27:8b:0d:9d
[ 5.441689] e1000 0000:00:08.0 eth1: Intel(R) PRO/1000 Network
Connection
[ 7.836309] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow
Control: RX
[ 7.854232] e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow
Control: RX
```

Ejercicio

1. Cree un directorio LKM en su directorio de base y acceda a él.
2. Escriba el siguiente código:

```
#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("John Doe");
MODULE_DESCRIPTION("Module hello world");
MODULE_VERSION("Version 1.00");

int init_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función
init_module().\n");
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función
cleanup_module().\n");
}
```

3. Escriba el archivo Makefile:

```
obj-m += hello.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD)
modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

install:
    cp ./hello.ko /lib/modules/$(shell uname
-r)/kernel/drivers/misc
```

4. Compile y luego instale.
5. Cargue el módulo de forma automática al arrancar el sistema.
6. Compruebe la carga tras el reinicio del sistema.
7. Muestre por pantalla la información del módulo.
8. Descargue el módulo y compruebe que el módulo hello se ha descargado correctamente.

procs

1. Presentación

El pseudo-sistema de archivos procs (process File System) se actualiza dinámicamente por el kernel. Se monta para que esté accesible en la carpeta /proc. Aunque no esté vacío, muestra un tamaño de 0 KB:

```
# ls -dl /proc
dr-xr-xr-x 110 root root 0 jul. 28 16:14 /proc
```

En efecto, el concepto del tamaño de archivo no tiene sentido para procs puesto que ocupa solamente una cantidad limitada de RAM.

El directorio /proc se utiliza para acceder a la información del kernel en ejecución. De esta forma podemos consultar en tiempo real información sobre el hardware, el sistema, la red, los procesos, etc. También es posible cambiar la configuración del kernel escribiendo en algunos archivos almacenados en /proc.

2. Principales archivos de /proc

/proc se organiza en carpetas y subcarpetas virtuales y agrupa los archivos por temas similares:

```
# ls /proc
1      18      2430    3196    459      dri      mtrr
10     19      2704    3200    5         driver  net
11     2       285     3203    6         execdomains pagetypeinfo
115    20      2889    3208    649      fb       partitions
12     2030   2918    3211    651      filesystems sched_debug
123    2085   2926    3212    7        fs      self
13     21     2927    3220    8        interrupts slabinfo
139    2117   2928    3223    9        iomem   softirqs
1394   2158   2929    3228    93       ioports stat
1395   2187   2930    3229    94       irq     swaps
1396   2206   2931    3314    95       kallsyms sys
1397   2224   2932    3315    acpi     kcore   sysrq-trigger
14     2248   2933    3322    asound   keys    sysvipc
140    2249   2936    361     buddyinfo key-users timer_list
15     2250   2940    378     bus      kmsg    timer_stats
16     2263   3      396     cgroups  kpagecount tty
1681   2287   3014    4       cmdline  kpageflags uptime
17     2290   3018    412     consoles loadavg  version
1724   2291   3025    415     cpuinfo  locks   vmallocinfo
1730   2293   3071    416     crypto   meminfo vmstat
1735   2313   3137    422     devices  misc    zoneinfo
1752   2321   3190    423     diskstats modules
1760   2394   3192    424     dma      mounts
```

Las carpetas que llevan un nombre numérico corresponden a un PID (Process identifier) de un proceso en ejecución. Las otras carpetas aportan información del hardware o del software.

a. /proc/acpi

El archivo `/proc/acpi/wakeup/` muestra los dispositivos que son capaces de arrancar un sistema que ha estado en espera prolongada. La columna `Device` enumera los dispositivos afectados mientras que la columna `Sysfs node` muestra el bus de conexión. La columna `Status`, por su parte, muestra el estado de funcionamiento de estos últimos, activado (enabled) o desactivado (disabled).

He aquí el contenido del archivo:

```
# cat /proc/acpi/wakeup
```

Device	S-state	Status	Sysfs node
PCI0	S5	*disabled	no-bus:pci0000:00
PCIE	S4	*disabled	pci:0000:00:1e.0
EHC2	S0	*enabled	pci:0000:00:1a.0
EHCI	S0	*enabled	pci:0000:00:1d.0
AZAL	S3	*disabled	pci:0000:00:1b.0
RP01	S3	*disabled	pci:0000:00:1c.0
RP02	S4	*disabled	pci:0000:00:1c.1
RP03	S3	*disabled	
RP04	S3	*disabled	pci:0000:00:1c.3
RP05	S3	*disabled	pci:0000:00:1c.4
RP06	S5	*disabled	pci:0000:00:1c.5
RP07	S5	*disabled	pci:0000:00:1c.6
RP08	S3	*disabled	pci:0000:00:1c.7
LID	S3	*disabled	
PBTN	S4	*disabled	

El archivo `/proc/acpi/sleep` se sustituye por otra interfaz en el pseudo-sistema de archivos `sysfs: /sys/power/state`.

b. /proc/apm

Este archivo proporciona información sobre el estado de la APM (Advanced Power Management) si la constante `CONFIG_APM` se ha declarado durante la compilación del kernel de Linux. Estos datos los utilizan los comandos APM.

c. /proc/bus

/proc/bus/pci

Esta carpeta almacena información sobre los buses PCI:

```
# ls -1 /proc/bus/pci
00
01
03
09
0c
10
3f
devices
```

/proc/bus/pci/devices proporciona información sobre los dispositivos PCI:

```
# cat /proc/bus/pci/devices
```

```
0000 80861237      0          0          0
0          0          0          0
0          0          0          0
0          0          0          0
0008 80867000      0          0          0
0          0          0          0
0          0          0          0
0          0          0          0
0009 80867111      0          1f0         3f6
170          376         d001          0
0          8          0          8
0          10         0          0
      ata_piix
0010 80eebeef      12         e0000008      0
0          0          0          0
2          1000000      0          0
0          0          0          0
0018 8086100e      13         f0000000      0
d011          0          0          0
0          20000      0          8
0          0          0          0
e1000
0020 80eecafe      14         d021         f0400000
f0800008          0          0
0          0          20         400000
4000          0          0          0
0
0028 80862415      15         d101         d201
0          0          0          0
0          100         40          0
0          0          0          0
```

snd_intel8x0				
0030	106b003f	16	f0804000	0
0		0	0	0
0		1000	0	0
0		0	0	0
ohci_hcd				
0038	80867113	9	0	0
0		0	0	0
0		0	0	0
0		0	0	0
0058	8086265c	13	f0805000	0
0		0	0	0
0		1000	0	0
0		0	0	0
ehci_hcd				
0068	80862829	15	d241	0
d251		0	d261	f0806000
0		8	0	8
0		10	2000	0
ahci				

/proc/bus/usb

Las distribuciones como Debian y Ubuntu descactivan usbfs y por lo tanto la carpeta /proc/bus/usb no está presente.

En cambio, CentOS tiene esta carpeta.

```
# ls -l /proc/bus/usb
total 0
dr-xr-xr-x. 2 root root 0 11 ago 16:25 001
dr-xr-xr-x. 2 root root 0 11 ago 16:25 002
-r--r--r--. 1 root root 0 11 ago 16:25 devices
```

El archivo `/proc/bus/usb/devices` contiene información sobre los buses:

```
# cat /proc/bus/usb/devices

T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 8
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P: Vendor=1d6b ProdID=0001 Rev= 2.06
S: Manufacturer=Linux 2.6.32-358.14.1.el6.x86_64 ohci_hcd
S: Product=OHCI Host Controller
S: SerialNumber=0000:00:06.0
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 2 Iv1=255ms

T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=480 MxCh= 8
B: Alloc= 0/800 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P: Vendor=1d6b ProdID=0002 Rev= 2.06
S: Manufacturer=Linux 2.6.32-358.14.1.el6.x86_64 ehci_hcd
S: Product=EHCI Host Controller
S: SerialNumber=0000:00:0b.0
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 4 Iv1=256ms
```

d. `/proc/cmdline`

Este archivo contiene los parámetros que se pasaron al kernel en el momento de su carga:

```
# cat /proc/cmdline

BOOT_IMAGE=/boot/vmlinuz-3.2.0-4-amd64 root=/dev/mapper/SystemVG-
rootLV ro quiet
```

e. /proc/cpuinfo

Este archivo contiene las características del o de los procesadores del equipo:

```
# cat cpuinfo
processor          : 0
vendor_id         : GenuineIntel
cpu family       : 6
model            : 30
model name       : Intel(R) Core(TM) i7 CPU           Q 740  @ 1.73GHz
stepping        : 5
cpu MHz         : 1723.555
cache size      : 6144 KB
physical id     : 0
siblings       : 2
core id        : 0
cpu cores      : 2
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 5
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 ht syscall nx
rdtscp lm constant_tsc rep_good nopl pni ssse3 lahf_lm
bogomips       : 3447.11
clflush size   : 64
cache_alignment : 64
address sizes  : 36 bits physical, 48 bits virtual
power management :

processor       : 1
vendor_id      : GenuineIntel
```

```
cpu family      : 6
model           : 30
model name      : Intel(R) Core(TM) i7 CPU          Q 740  @ 1.73GHz
stepping        : 5
cpu MHz         : 1723.555
cache size      : 6144 KB
physical id     : 0
siblings        : 2
core id         : 1
cpu cores       : 2
apicid          : 1
initial apicid  : 1
fpu             : yes
fpu_exception   : yes
cpuid level     : 5
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 ht syscall nx
rdtscp lm constant_tsc rep_good nopl pni ssse3 lahf_lm
bogomips        : 3451.59
clflush size    : 64
cache_alignment : 64
address sizes    : 36 bits physical, 48 bits virtual
power management :
```

f. /proc/crypto

Este archivo muestra todos los cifrados criptográficos instalados y utilizados:

```
# cat /proc/crypto
name      : stdrng
driver    : krng
module    : kernel
priority  : 200
refcnt    : 1
selftest  : passed
type      : rng
seedsize  : 0

name      : md5
driver    : md5-generic
module    : kernel
priority  : 0
refcnt    : 1
selftest  : passed
type      : shash
blocksize : 64
digestsize : 16
```

g. /proc/devices

Este archivo contiene los dispositivos de bloques y de caracteres que se cargan en el sistema.

```
# cat devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttyS
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 6 lp
 7 vcs
10 misc
13 input
21 sg
29 fb
99 ppdev
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
216 rfcomm
226 drm
252 hidraw
253 bsg
254 rtc

Block devices:
259 blkext
```



```
7 loop
8 sd
11 sr
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
254 device-mapper
```

El número situado delante del nombre del dispositivo es el número principal que identifica el tipo de dispositivo.

h. /proc/dma

Este archivo contiene la lista de canales DMA ISA que están en uso:

```
# cat /proc/dma
4: cascade
```

i. /proc/filesystems

Este archivo muestra los sistemas de archivos que tiene en cuenta el kernel:

```
# cat /proc/filesystems
```

```
nodev    sysfs
nodev    rootfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    tmpfs
nodev    devtmpfs
nodev    debugfs
nodev    securityfs
nodev    ockfs
nodev    pipefs
nodev    anon_inodefs
nodev    devpts
nodev    ramfs
nodev    hugetlbfs
nodev    pstore
nodev    mqueue
          ext4
nodev    usbfs
nodev    rpc_pipefs
nodev    nfs
nodev    nfs4
nodev    nfsd
nodev    binfmt_misc
```

j. /proc/fs

Esta carpeta contiene subcarpetas relativas a los sistemas de archivos:

Archivo	Descripción
ext4	Información sobre el sistema de archivos montado ext4.
fscache/stats	Datos estadísticos sobre fscache, que es una caché local persistente. La utilizan los sistemas de archivo para tomar los datos procedentes de la red y ponerlos en caché en el disco local. Esto permite reducir el tráfico de red para los usuarios que acceden a los datos de un sistema de archivos montado en red.
jbd2	Información y estadísticas sobre el sistema de archivos ext4.
nfs	Información sobre los sistemas de archivos que se están exportando a un cliente.
nfsd	Información sobre el servidor NFS.
nfsfs	Información sobre el sistema de archivos NFS.

k. /proc/ide

Si el sistema tiene un bus IDE entonces existe esta carpeta. Contiene un árbol de carpetas para cada canal IDE y los dispositivos conectados.

l. /proc/interrupts

Este archivo contiene el número de interrupciones por IRQ en la arquitectura x86:

```
# cat /proc/interrupts

          CPU0       CPU1
0:         45         0    IO-APIC-edge    timer
1:          9         0    IO-APIC-edge    i8042
8:          0         0    IO-APIC-edge    rtc0
9:          0         0    IO-APIC-fasteoi  acpi
12:        146         0    IO-APIC-edge    i8042
14:          0         0    IO-APIC-edge    ata_piix
15:         92         0    IO-APIC-edge    ata_piix
19:       39213         0    IO-APIC-fasteoi  ehci_hcd:usb1, eth0
20:        4646         0    IO-APIC-fasteoi  vboxguest
21:       18410         0    IO-APIC-fasteoi  ahci, snd_intel8x0
22:         26         0    IO-APIC-fasteoi  ohci_hcd:usb2
NMI:         0         0    Non-maskable interrupts
LOC:    138073    137387    Local timer interrupts
SPU:         0         0    Spurious interrupts
PMI:         0         0    Performance monitoring interrupts
IWI:         0         0    IRQ work interrupts
RES:       27938     20542    Rescheduling interrupts
CAL:        217        428    Function call interrupts
TLB:        828       1247    TLB shootdowns
TRM:         0         0    Thermal event interrupts
THR:         0         0    Threshold APIC interrupts
MCE:         0         0    Machine check exceptions
MCP:        153        153    Machine check polls
ERR:         0
MIS:         0
```

La primera columna indica el número de IRQ. A partir de la versión del kernel Linux 2.6.24 para las arquitecturas i386 y x86_64 incluye las interrupciones internas del sistema que no se conectan directamente a un dispositivo tales como los NMI (Non Maskable Interrupt), LOC (Local Timer Interrupt), CAL (Remote Function Call Interrupt), etc. A estas últimas no tenemos acceso.

Cada unidad central (CPU) posee su propia columna y su propio número de interrupciones por IRQ.

La penúltima columna muestra el tipo de interrupción y la última columna contiene el nombre del dispositivo que utiliza esa IRQ.

m. /proc/iomem

Este archivo muestra la memoria en el sistema para cada dispositivo físico:

```
# cat /proc/iomem
00000000-0000ffff : reserved
00010000-0009ffff : System RAM
0009f000-0009ffff : reserved
000a0000-000bffff : PCI Bus 0000:00
000c0000-000cf7ff : Video ROM
000cf800-000cffff : Adapter ROM

...

fee10000-ff9fffff : PCI Bus 0000:00
ffa00000-ffbfffff : pnp 00:0b
ffc00000-ffdfffff : PCI Bus 0000:00
ffe00000-ffffffff : reserved
    ffe00000-ffffffff : pnp 00:0b
100000000-21bfffffff : System RAM
```

n. /proc/ioprots

Este archivo contiene información sobre los puertos de E/S utilizados por el sistema:

```
# cat /proc/ioprots
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-0060 : keyboard
0064-0064 : keyboard
0070-0071 : rtc_cmos
    0070-0071 : rtc0
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:01.1
    0170-0177 : ata_piix
01f0-01f7 : 0000:00:01.1
    01f0-01f7 : ata_piix
0376-0376 : 0000:00:01.1
    0376-0376 : ata_piix
03c0-03df : vga+
03f6-03f6 : 0000:00:01.1
    03f6-03f6 : ata_piix
0cf8-0cff : PCI conf1
4000-4003 : ACPI PM1a_EVT_BLK
4004-4005 : ACPI PM1a_CNT_BLK
4008-400b : ACPI PM_TMR
4020-4021 : ACPI GPE0_BLK
d000-d00f : 0000:00:01.1
    d000-d00f : ata_piix
d010-d017 : 0000:00:03.0
```

```
d010-d017 : e1000
d020-d03f : 0000:00:04.0
d100-d1ff : 0000:00:05.0
    d100-d1ff : Intel 82801AA-ICH
d200-d23f : 0000:00:05.0
    d200-d23f : Intel 82801AA-ICH
d240-d247 : 0000:00:0d.0
    d240-d247 : ahci
d250-d257 : 0000:00:0d.0
    d250-d257 : ahci
d260-d26f : 0000:00:0d.0
    d260-d26f : ahci
```

o. /proc/irq

Esta carpeta cuenta con una subcarpeta por IRQ:

```
# ls -l /proc/irq
total 0
dr-xr-xr-x 2 root root 0 abr 30 14:27 0
dr-xr-xr-x 3 root root 0 abr 30 14:27 1
dr-xr-xr-x 2 root root 0 abr 30 14:27 10
dr-xr-xr-x 2 root root 0 abr 30 14:27 11
dr-xr-xr-x 3 root root 0 abr 30 14:27 12
dr-xr-xr-x 2 root root 0 abr 30 14:27 13
dr-xr-xr-x 3 root root 0 abr 30 14:27 14
dr-xr-xr-x 3 root root 0 abr 30 14:27 15
dr-xr-xr-x 3 root root 0 abr 30 14:27 19
dr-xr-xr-x 2 root root 0 abr 30 14:27 2
dr-xr-xr-x 3 root root 0 abr 30 14:27 20
dr-xr-xr-x 4 root root 0 abr 30 14:27 21
dr-xr-xr-x 3 root root 0 abr 30 14:27 22
dr-xr-xr-x 2 root root 0 abr 30 14:27 3

dr-xr-xr-x 2 root root 0 abr 30 14:27 4
dr-xr-xr-x 2 root root 0 abr 30 14:27 5
dr-xr-xr-x 2 root root 0 abr 30 14:27 6
dr-xr-xr-x 2 root root 0 abr 30 14:27 7
dr-xr-xr-x 3 root root 0 abr 30 14:27 8
dr-xr-xr-x 3 root root 0 abr 30 14:27 9
-rw-r--r-- 1 root root 0 abr 30 14:27 default_smp_affinity
```


/proc/irq/default_smp_affinity define de forma global el valor por defecto del SMP affinity que permite elegir en qué procesador efectuar las operaciones. Este valor es una máscara binaria. Imaginemos que contamos con un equipo dotado de cuatro procesadores:

CPU	Binario	Hexadecimal
CPU 0	0001	1
CPU 1	0010	2
CPU 2	0100	4
CPU 3	1000	8

Si deseamos, por ejemplo, el procesamiento en la CPU 0 y CPU 2, debemos sumar los dos valores binarios 0001 + 0100, lo que nos da: 0101 (5 en hexadecimal).

Para un equipo bi-procesador, /proc/irq/default_smp_affinity contiene el valor hexadecimal 3 (es decir 0001+0010=0011):

```
# cat default_smp_affinity
3
```

En cada carpeta de una IRQ, tenemos un archivo smp_affinity que especifica, de forma respectiva, la elección del procesador para dicha interrupción.

Ejemplo

El objetivo es cambiar el SMP affinity de IRQ de la tarjeta de red eth0 a 2, para utilizar la CPU 1.

¿Cuál es su IRQ?

```
# grep eth0 /proc/interrupts
19:      41133          0   IO-APIC-fastEOI    ehci_hcd:usb1, eth0
```

Es la IRQ 19. Basta con modificar el archivo /proc/irq/19/smp_affinity:

```
# echo "2" > /proc/irq/19/smp_affinity
```

Podemos comprobar visualizando varias veces el archivo `/proc/interrupts` que el valor de la IRQ 19 aumentará en la columna de la CPU1:

```
# grep 19 /proc/interrupts
19:      52586      842  IO-APIC-fastEOI  ehci_hcd:usb1, eth0
# grep 19 /proc/interrupts
19:      52586      854  IO-APIC-fastEOI  ehci_hcd:usb1, eth0
# grep 19 /proc/interrupts
19:      52586      934  IO-APIC-fastEOI  ehci_hcd:usb1, eth0
```

El valor de la CPU 0 no evoluciona.

p. `/proc/kcore`

Este pseudo-archivo representa la memoria física del sistema:

```
# ls -l kcore
-r----- 1 root root 140737477877760 abr 30 14:31 kcore
```

La longitud total de este archivo es el tamaño de la memoria RAM física más 4 KB. En los kernels de 64 bits, el tamaño de este archivo puede alcanzar 128 terabytes.

q. `/proc/loadavg`

Este archivo contiene información sobre la carga del sistema:

```
# cat /proc/loadavg
0.01 0.20 0.21 1/356 2713
```

Los tres primeros valores representan el número de tareas activas en el sistema como los procesos en ejecución, por ejemplo, con una media de los 1, 5 y 15 últimos minutos.

El siguiente valor indica el número actual de tareas ejecutables como los procesos planificados y el número total de procesos en el sistema.

El último valor corresponde al PID del proceso activo más reciente.

r. /proc/mdstat

Este archivo contiene una lista de los volúmenes RAID (Redundant Array of Independent Disks):

```
#cat /proc/mdstat
Personalities : [Raid 1]
md2  : active raid1 sdb2[1] sda2[0]
      40957232 blocks [2/2] [UU]
md1  : active raid1 sdb3[1] sda3[0]
      8385856 blocks [2/2] [UU]
md3  : active raid1 sdb5[1] sda5[0]
      194563072 blocks [2/2] [UU]
md0  : active raid1 sdb1[1] sda1[0]
      104320 blocks [2/2] [UU]
unused devices : <none>
```

s. /proc/meminfo

Este archivo nos permite ver el uso de la memoria:

```
# cat /proc/meminfo
MemTotal:          508780 kB
MemFree:           54628 kB
Buffers:           99380 kB
Cached:            208556 kB
SwapCached:         0 kB
Active:            177796 kB
Inactive:          199132 kB
Active(anon):       69480 kB
Inactive(anon):     212 kB
Active(file):       108316 kB
Inactive(file):     198920 kB
Unevictable:        0 kB
Mlocked:            0 kB
SwapTotal:         499708 kB
SwapFree:          499708 kB
Dirty:              0 kB
Writeback:          0 kB
AnonPages:          69008 kB
Mapped:             33920 kB
Shmem:              708 kB
Slab:               59392 kB
SReclaimable:       47800 kB
SUnreclaim:         11592 kB
KernelStack:        1688 kB
PageTables:         7512 kB
NFS_Unstable:        0 kB
Bounce:             0 kB
WritebackTmp:        0 kB
CommitLimit:       754096 kB
```

```
Committed_AS:      427788 kB
VmallocTotal:    34359738367 kB
VmallocUsed:      18164 kB
VmallocChunk:    34359719299 kB
HardwareCorrupted:    0 kB
AnonHugePages:      0 kB
HugePages_Total:    0
HugePages_Free:     0
HugePages_Rsvd:     0
HugePages_Surp:     0
Hugepagesize:      2048 kB
DirectMap4k:       36800 kB
DirectMap2M:       487424 kB
```

Significado de los principales valores:

Valor	Descripción
MemTotal	Total de RAM disponible. La RAM disponible es la RAM física menos algunos bits reservados y el código binario del kernel.
MemFree	Cantidad de RAM no asignada actualmente.
Buffers	Cantidad de RAM asignada actualmente a los buffers.
Cached	Cantidad de memoria utilizada actualmente como caché.
SwapCached	Cantidad de memoria en proceso de salida del swap. Sin embargo, una copia siempre está presente en el swap.
Active	Cantidad de memoria que se ha utilizado hace poco y que no suele recuperarse, salvo si es absolutamente necesario.
Inactive	Cantidad de memoria que no se ha utilizado desde hace algún tiempo.

SwapTotal	Tamaño del swap.
SwapFree	Espacio disponible en el swap.
Dirty	Cantidad de memoria en espera para ser escrita al disco.
Writeback	Cantidad de memoria que se encuentra en proceso de escritura al disco.
Slab	Cantidad de memoria utilizada en el kernel y asignada a Slab para sus propios objetos y estructuras.
PageTables	Cantidad de memoria utilizada como tabla de páginas para gestionar la paginación.
CommitLimit	Estimación de la cantidad máxima de memoria disponible para ser asignada.
VmallocTotal	Cantidad total de memoria del espacio de direcciones virtuales del kernel.
VmallocUsed	Cantidad de memoria utilizada por el espacio de direcciones virtuales del kernel.
VmallocChunk	El mayor bloque contiguo de memoria virtual disponible en el espacio de direcciones virtuales del kernel.
Hugepagesize	Tamaño de una hugepage (2048 KB por defecto). Es una página de gran tamaño usada por programas que tienen necesidades específicas de memoria. El número de páginas hugepage se define en el siguiente archivo: /proc/sys/vm/nr_shugepages (el valor por defecto es 0).
HugePages_Total	Número de hugepages asignadas.
HugePages_Free	Número de hugepages disponibles.



Cabe señalar que la información hugepages opera solo en las arquitecturas x86, Itanium y AMD64.

t. /proc/misc

Este archivo contiene un resumen de diversos dispositivos:

```
# cat /proc/misc
57 rfkill
223 uinput
237 loop-control
58 vboxuser
59 vboxguest
236 device-mapper
60 network_throughput
61 network_latency
62 cpu_dma_latency
1 psaux
228 hpet
231 snapshot
227 mcelog
63 vga_arbiter
```

u. /proc/modules

Este archivo muestra una lista de los módulos cargados por el kernel:

```
# cat /proc/modules
vboxvideo 12437 1 - Live 0xfffffffffa0369000 (0)
drm 183952 2 vboxvideo, Live 0xfffffffffa03fc000
ppdev 12763 0 - Live 0xfffffffffa0364000
lp 17149 0 - Live 0xfffffffffa035e000
...

pcspkr 12579 0 - Live 0xfffffffffa0100000
```

```
vboxguest 148611 2 - Live 0xfffffffffa011b000 (0)
thermal_sys 18040 1 processor, Live 0xfffffffffa00fa000
button 12937 0 - Live 0xfffffffffa0116000
ext4 350763 3 - Live 0xfffffffffa00a3000
crc16 12343 2 bluetooth,ext4, Live 0xfffffffffa008b000
mbcache 13114 1 ext4, Live 0xfffffffffa0061000
jbd2 62115 1 ext4, Live 0xfffffffffa0092000
dm_mod 63645 12 - Live 0xfffffffffa0076000
sd_mod 36136 2 - Live 0xfffffffffa006c000
crc_t10dif 12348 1 sd_mod, Live 0xfffffffffa0067000
ahci 24997 1 - Live 0xfffffffffa0034000
libahci 22860 1 ahci, Live 0xfffffffffa0029000
libata 140630 4 ata_generic,ata_piix,ahci,libahci, Live
0xfffffffffa003d000
scsi_mod 162269 4 sg,sr_mod,sd_mod,libata, Live 0xfffffffffa0000000
```

v. /proc/mounts

Es un enlace simbólico que apunta a self/mounts.

```
# ls -l /proc/mounts
lrwxrwxrwx 1 root root 11 abr 30 14:51 /proc/mounts ->
self/mounts
```

Contiene la tabla de montaje:

```
# cat mounts
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs
rw,relatime,size=10240k,nr_inodes=62945,mode=755 0 0
devpts /dev/pts devpts
rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0

tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=50880k,mode=755 0 0
```



```

/dev/mapper/SystemVG-rootLV / ext4 rw,relatime,errors=remount-
ro,user_xattr,barrier=1,data=ordered 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
tmpfs /run/shm tmpfs rw,nosuid,nodev,noexec,relatime,size=101740k 0 0
/dev/mapper/SystemVG-homeLV /home ext4
rw,relatime,user_xattr,barrier=1,data=ordered 0 0
/dev/mapper/SystemVG-optLV /opt ext4
rw,relatime,user_xattr,barrier=1,data=ordered 0 0
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc
rw,nosuid,nodev,noexec,relatime 0 0

```

w. /proc/net

El enlace simbólico /proc/net apunta a /proc/self/net que contiene información variada sobre las funcionalidades de red.

Archivo	Descripción
arp	Almacena la tabla ARP para la resolución de direcciones. Utilizado por el comando arp.
rarp	Búsqueda inversa de ARP. Este archivo existe si RARP se configura en el kernel. Utilizado por el comando rarp.
dev	Estado de los dispositivos de red para conocer el número de paquetes emitidos y recibidos, el número de errores y de colisiones, etc. Utilizado por el comando ifconfig.
igmp	Información sobre el multicast.
snmp	Información sobre los protocolos IP, ICMP, TCP y UDP para un agente SNMP.
tcp	Tabla de sockets TCP.
udp	Tabla de sockets UDP.

unix	Lista de sockets en el dominio UNIX.
netstat	Datos utilizados por el comando netstat.
route	Tabla de enrutamiento. Utilizado por el comando route.
wireless	Información sobre el estado de la red inalámbrica.

x. /proc/partitions

Este archivo proporciona información sobre el particionado de los discos.

```
# cat /proc/partitions
major minor #blocks name

 8         0  52428800 sda
 8         1  52426752 sda1
254        0  14647296 dm-0
254        1    499712 dm-1
 11        0   1048575 sr0
254        2  37105664 dm-2
254        3   172032 dm-3
```

El número mayor identifica el controlador de dispositivo, mientras que el número menor identifica el dispositivo. La tercera columna muestra los bloques y la última columna el nombre que está montado en /dev.

y. /proc/self

Un enlace simbólico self señala al PID del proceso actual, en este caso el PID es 3525:

```
#ls -l /proc/self
lrwxrwxrwx 1 root root 0 abr 30 14:23 /proc/self -> 3525
```

z. /proc/stat

Este archivo ofrece también estadísticas que se remontan al último arranque del sistema.

```
# cat /proc/stat  
cpu 3731 1 9436 28164682 6485 0 727 0 0 0  
cpu0 3019 1 5683 14074382 5633 0 546 0 0 0  
cpu1 711 0 3753 14090299 851 0 180 0 0 0  
intr 1128014 45 9 0 0 0 0 0 0 0 0 0 0 154 0 0 92 0 0 0 146472  
14404 19774 26 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
ctxt 2095858  
btime 1375570816  
processes 4954  
procs_running 2  
procs_blocked 0  
softirq 722318 0 219820 28209 144840 17904 0 2 180470 3873 127200
```

aa. /proc/swaps

Este archivo contiene información acerca de la partición de swap:

```
# cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/dm-1	partition	499708	0	-1

ab. /proc/sys

/proc/sys/dev

/proc/sys/dev/cdrom/info contiene información sobre el lector de CDROM:

```
# cat /proc/sys/dev/cdrom/info
```

CD-ROM information, Id: cdrom.c 3.20 2003/12/17

drive name:	sr0
drive speed:	32
drive # of slots:	1
Can close tray:	1
Can open tray:	1
Can lock tray:	1
Can change speed:	1
Can select disk:	0
Can read multisession:	1
Can read MCN:	1
Reports media changed:	1
Can play audio:	1
Can write CD-R:	0
Can write CD-RW:	0
Can read DVD:	1
Can write DVD-R:	0
Can write DVD-RAM:	0
Can read MRW:	1
Can write MRW:	1
Can write RAM:	1

/proc/sys/dev/parport almacena información de los puertos paralelos. En su interior habrá una carpeta para cada puerto. parport0, parport1...

```
# ls -l /proc/sys/dev/parport/partport0
autoprobe
autoprobe0
autoprobe1
autoprobe2
autoprobe3
devices
base-addr
irq
dma
modes
spintime
```

/proc/sys/fs

/proc/sys/fs contiene la configuración de los sistemas de archivos.

Archivo	Descripción
file-max	Límite del sistema del número de archivos abiertos por un proceso. El aumento de file-max requiere el aumento del número de inodos en el archivo /proc/sys/fs/inode-max.
inode-max	Número máximo de inodos en memoria. Este valor debe ser 3 a 4 veces mayor que el número file-max derivado del hecho de que los descriptores STDIN STDOUT, y también los sockets de red necesitan inodos. En caso de falta de inodos, debemos incrementar el valor.
inode-nr	El primer valor (nr_inodes) es el número de inodos asignados por el sistema que puede ser ligeramente mayor que inode-max, ya que Linux los asigna por página completa. El segundo valor (nr_free_inodes) representa el número de inodos disponibles.

inode-state	Contiene siete campos: • Los dos primeros valores son los de inode-nr. • El tercer valor (preshrink) no es nulo cuando nr_inodes > inode-max y la lista de inodes debe ser purgada en lugar de asignar más. • Los otros cuatro valores no son significativos.
fs/file-nr	Número de archivos abiertos actualmente. Contiene tres valores: • El número de descriptores de archivos asignados, • El número de descriptores de archivos libres, • El número máximo de descriptores de archivos libres.
lease-break-time	Tiempo de gracia que el kernel de Linux asigna a un proceso que posea un bloqueo de archivo después de haberle enviado una señal que indica que otro proceso espera para abrir el archivo. Si durante el período de gracia el titular del bloqueo no lo elimina o no disminuye su alcance, entonces Linux eliminará el bloqueo por la fuerza.
leases-enable	Activa o inhibe los bloqueos de archivos para todo el sistema. El valor 0 desactiva los bloqueos mientras que un valor no nulo los activa.
protected_hardlinks	Cuando el valor de este archivo es 0, ninguna restricción impide la creación de vínculos físicos. Este era el caso antes del kernel Linux 3.6. Cuando el valor es 1, es posible crear un vínculo físico a un archivo objetivo siempre que: • el llamante cuente con la capacidad de evitar las verificaciones de permisos en las operaciones que requieren normalmente que el UID del proceso se corresponda con el UID del archivo (CAP_FOWNER); • el UID del proceso que crea el enlace se corresponda con el UID del propietario del archivo de destino; • el objetivo sea un archivo regular;

	• el objetivo no cuente con el bit de permisos Set-UID activado; • el objetivo no cuente con ninguno de sus bits de permisos Set-GID y ejecutable por el grupo activado; • el llamante cuente con el permiso de leer y escribir en el archivo correspondiente.
protected_symlinks	Cuando el valor de este archivo es 0, ninguna restricción impide la creación y el seguimiento de enlaces físicos. Este era el caso antes del kernel Linux 3.6. Cuando el valor es 1, los enlaces simbólicos no se siguen para determinadas circunstancias: • el UID del proceso que sigue al enlace corresponde al del propietario del enlace simbólico; • el enlace no es en un directorio accesible en escritura para todos; • el enlace simbólico y su directorio padre tienen el mismo propietario.

/proc/sys/kernel

/proc/sys/kernel contiene varios archivos de configuración que afectan directamente al funcionamiento del kernel Linux:

```
# ls /proc/sys/kernel
acct                perf_event_max_sample_rate
acpi_video_flags    perf_event_mlock_kb
auto_msgmni         perf_event_paranoid
blk_iopoll          pid_max
bootloader_type     poweroff_cmd
bootloader_version  print-fatal-signals
cad_pid            printk
cap_last_cap        printk_delay
compat-log          printk_ratelimit
core_pattern        printk_ratelimit_burst
core_pipe_limit     pty
core_uses_pid       random
```

ctrl-alt-del	randomize_va_space
dmesg_restrict	real-root-dev
domainname	sched_autogroup_enabled
ftrace_dump_on_oops	sched_child_runs_first
hostname	sched_domain
hotplug	sched_latency_ns
hung_task_check_count	sched_migration_cost
hung_task_panic	sched_min_granularity_ns
hung_task_timeout_secs	sched_nr_migrate
hung_task_warnings	sched_rt_period_us
io_delay_type	sched_rt_runtime_us
keys	sched_shares_window
kptr_restrict	sched_time_avg
kstack_depth_to_print	sched_tunable_scaling
max_lock_depth	sched_wakeup_granularity_ns
modprobe	sem
modules_disabled	shmall
msgmax	shmmax
msgmnb	shmmni
msgmni	shm_rmid_forced
ngroups_max	softlockup_panic
nmi_watchdog	sysrq
osrelease	tainted
ostype	threads-max
overflowgid	timer_migration
overflowuid	unknown_nmi_panic
panic	usermodehelper
panic_on_io_nmi	version
panic_on_oops	watchdog
panic_on_unrecovered_nmi	watchdog_thresh

He aquí un cuadro de los principales archivos `/proc/sys/kernel`:

Archivo	Descripción
acct	Este archivo contiene tres valores que controlan la contabilidad de procesos. Los valores por defecto son 4, 2 y 30. Esto significa que la contabilidad se suspende por debajo del 2% de espacio libre y luego se reanuda a partir del 4%. La verificación se realiza cada 30 segundos.
ctrl-alt-supr	Control de la gestión de la combinación de teclas [Ctrl][Alt][Supr] del teclado. Si el archivo contiene el valor 0 al capturar [Ctrl][Alt][Supr] se envía al programa init para reiniciar el sistema correctamente. Si el valor es superior a 0, Linux provoca el reinicio inmediato.
hostname	Contiene el nombre del host.
domainname	Contiene el nombre del dominio DNS.
modprobe	Contiene la ruta del cargador de módulos del kernel. Por defecto, su valor es <code>/sbin/modprobe</code>. Este archivo solo existe si el kernel está compilado con la opción <code>CONFIG_MODULES</code> activada.
modules_disabled	Activa o desactiva la carga de los módulos del kernel. El valor por defecto es 0. Si el valor es 1 entonces los módulos no podrán ser cargados o descargados. Este valor no puede redefinirse a 0. Este archivo solo existe si el kernel está compilado con la opción <code>CONFIG_MODULES</code> activada.
osrelease	Versión del kernel de Linux.
ostype	Tipo de sistema, es decir, Linux.
version	Contiene una cadena que difiere de una distribución a otra.

	Debian: #1 SMP Debian 3.2.46-1 Ubuntu: #70-Ubuntu SMP Wed May 29 20:12:06 UTC 2013 CentOS: #1 SMP Tue Jul 16 23:51:20 UTC 2013 El campo "#1" indica el número de compilación del kernel desde estas fuentes y la fecha corresponde a la fecha de compilación. En el caso de Debian, no existe ninguna fecha.
panic	Accede en modo lectura y escritura a la variable panic_timeout del kernel. Si vale 0, el kernel se pondrá en bucle en caso de fallo. De lo contrario, indica que el kernel debe reiniciar por sí mismo después del número de segundos que contiene.
panic_on_oops	Controla el comportamiento del kernel de Linux cuando se produce un «Oops» (kernel panic) o un error. Si contiene 0, el sistema intenta seguir trabajando. Si contiene 1, el sistema espera unos segundos para dejar tiempo para grabar la salida del Oops en los registros y arranca un «pánico». Si /proc/sys/kernel/panic tiene un valor no nulo entonces el equipo reiniciará.
pid_max	Establece el número máximo de PID. El valor por defecto es 32768.
pty/max	Muestra el número máximo de pseudo-terminales. El valor por defecto es 4096.
pty/nr	Indica el número de pseudo-terminales utilizados actualmente.
sem	Establece el número máximo y el tamaño de los semáforos (recursos System V IPC).
shmall	Define la cantidad máxima de memoria compartida (recursos System V IPC).

shmmax	Define la cantidad máxima de un segmento de memoria compartida (recursos System V IPC).
shmmni	Número máximo de segmentos de memoria compartida (recursos System V IPC).
sysrq	El valor 0 desactiva sysrq (magic sysrq key), que permite mediante una combinación de teclas ejecutar comandos de bajo nivel. El valor 1 activa la funcionalidad.
threads-max	Define el número máximo de hilos de ejecución (threads) por proceso. El valor por defecto varía en función de la configuración del equipo.

/proc/sys/net

/proc/sys/net contiene a su vez subcarpetas relativas a la red:

```
# ls -l /proc/sys/net
total 0
dr-xr-xr-x 1 root root 0 abr 30 18:56 core
dr-xr-xr-x 1 root root 0 abr 30 18:56 ipv4
dr-xr-xr-x 1 root root 0 abr 30 14:23 ipv6
dr-xr-xr-x 1 root root 0 abr 30 18:56 netfilter
dr-xr-xr-x 1 root root 0 abr 30 18:56 token-ring
dr-xr-xr-x 1 root root 0 abr 30 14:22 unix
```

Pueden existir otras carpetas en función de nuestra configuración como Ethernet, ipx, bridging, x25, etc.

/proc/sys/net/core contiene los parámetros que controlan la interacción entre el kernel de Linux y las capas de red. Estos son los principales archivos:

Archivo	Descripción
message_burst	Define el tiempo necesario a la décima de segundo para escribir un nuevo mensaje de advertencia. Este parámetro se utiliza para mitigar los ataques por denegación de servicio (DoS). Por defecto, el valor es 10.

message_cost	Define un coste para cada mensaje de aviso. A mayor valor de este archivo, mayor probabilidad de que el mensaje de aviso sea ignorado. Este parámetro se utiliza también para mitigar los ataques DoS. Por defecto, el valor es 5.
netdev_max_backlog	Establece el número máximo de paquetes autorizados en la cola cuando una interfaz específica recibe paquetes más rápido de lo que el kernel puede procesar. El valor por defecto de este archivo es 300.
optmem_max	Define el tamaño máximo de los buffers auxiliares autorizados por socket.
rmem_default	Establece el tamaño por defecto de los buffers de recepción utilizados por los sockets.
rmem_max	Establece el tamaño máximo de los buffers de recepción utilizados por los sockets.
wmem_default	Establece el tamaño por defecto de los buffers de envío utilizados por los sockets.
wmem_max	Establece el tamaño máximo de los buffers de envío utilizados por los sockets.

/proc/sys/net/ipv4

/proc/sys/net/ipv4 contiene los archivos de configuración para TCP/IP versión 4:

Archivo	Descripción
icmp_echo_ignore_all	Si el valor es 1, entonces el kernel ignora todas las solicitudes ECHO ICMP que recibe. El valor por defecto de este archivo es 0.
icmp_echo_ignore_broadcasts	Si el valor es 1, entonces el kernel ignora todas las solicitudes ECHO ICMP ECHO y TIMESTAMP que se

	le envían por broadcast y multicast, de lo contrario, configurar un 0. El valor por defecto de este archivo es 1.
ip_default_ttl	Define el valor por defecto del campo TTL (Time to Live) para los paquetes IP salientes. El valor debe estar comprendido entre 1 y 255. Por defecto, es 64.
ip_forward	El valor 1 permite enviar los paquetes entre las interfaces, o bien 0 para desactivar. El valor por defecto de este archivo es 0.
ip_local_port_range	Define el rango de puertos TCP y UDP. El primer número vale por defecto 32768. El segundo número vale por defecto 61000.

/proc/sys/net/ipv6

/proc/sys/net/ipv6 contiene los archivos de configuración para TCP/IP versión 6.

Si el valor del archivo /proc/sys/net/ipv6/bindv6only es igual a 1 entonces la comunicación se establece solo a través de IPv6. Si el valor es 0 entonces la comunicación es mixta, IPv4 y IPv6. El valor por defecto es 0.

/proc/sys/net/ipv6/conf contiene un conjunto de subcarpetas:

```
# ls -l /proc/sys/net/ipv6/conf
total 0
dr-xr-xr-x 1 root root 0 abr 30 19:08 all
dr-xr-xr-x 1 root root 0 abr 30 19:08 default
dr-xr-xr-x 1 root root 0 abr 30 14:23 eth0
dr-xr-xr-x 1 root root 0 abr 30 19:08 lo
```

Permiten la configuración global, por defecto o para una interfaz:

Carpeta	Descripción
all	Configuración para todas las interfaces.
default	Configuración por defecto.
eth0	Configuración para la interfaz eth0. Existe una carpeta por interfaz.
lo (loopback)	Configuración para la interfaz loopback (127.0.0.1).

/proc/sys/net/ipv6/icmp contiene la configuración del protocolo ICMP:

```
# ls /proc/sys/net/ipv6/icmp
ratelimit
```

El archivo ratelimit limita el flujo del tráfico ICMPv6 enviado. El valor por defecto es 1000.

/proc/sys/net/ipv6/neigh contiene los parámetros por defecto para la detección del entorno cercano (vecinos) y también para cada interfaz.

```
# ls /proc/sys/net/ipv6/neigh
default  eth0  lo
```

/proc/sys/net/ipv6/route contiene los parámetros para el enrutamiento:

```
# ls -1 /proc/sys/net/ipv6/route
flush
gc_elasticity
gc_interval
gc_min_interval
gc_min_interval_ms
gc_thresh
gc_timeout
max_size
min_adv_mss
mtu_expires
```

/proc/sys/vm

/proc/sys/vm contiene los archivos de configuración de la gestión de memoria, los buffers y la cache.

Archivo	Descripción
drop_caches	El archivo contiene los valores siguientes: 1 para liberar la caché, 2 para liberar dentries e inodos, 3 para liberar pagecache, dentries e inodos.
legacy_va_layout	Si el valor no es 0 entonces la nueva mmap 32 bits se encuentra desactivada para todos los procesos.
memory_failure_early_kill	Determina cómo matar un proceso cuando se detecta un error de memoria no corregida en segundo plano por un hardware que no puede ser tratado por el kernel. El valor 1 mata todos los procesos que tienen la página de memoria corrupta y no recargable tan pronto como se detecta la corrupción. El valor 0 desasigna solo la página de memoria corrupta de todos los procesos y mata solo el proceso que intenta acceder a ella. La señal enviada es un SIGBUS (7).
memory_failure_recovery	Si la plataforma lo soporta, activa la recuperación de un error de memoria. Válido desde el kernel 2.6.32.
swappiness	Permite que el kernel favorezca el swap o bien la RAM. El valor puede estar situado entre 0 y 100. El valor por defecto es 60. Cuanto mayor sea el valor, el kernel favorecerá el uso del swap. Esto puede afectar al rendimiento.

ac. /proc/scsi

/proc/scsi presenta varios subdirectorios para los controladores que contienen, respectivamente, un archivo para cada host SCSI del sistema:

```
# ls -1 /proc/scsi/  
device_info  
scsi  
sg
```

/proc/scsi/scsi muestra todos los dispositivos SCSI reconocidos:

```
# cat /proc/scsi/scsi  
Attached devices:  
Host: scsi0 Channel: 00 Id: 00 Lun: 00  
  Vendor: ATA      Model: TOSHIBA MK5056GS Rev: LJ00  
  Type:   Direct-Access          ANSI  SCSI revision: 05  
Host: scsi1 Channel: 00 Id: 00 Lun: 00  
  Vendor: HL-DT-ST Model: DVD+-RW GS20N   Rev: A110  
  Type:   CD-ROM                ANSI  SCSI revision: 05  
Host: scsi2 Channel: 00 Id: 00 Lun: 00  
  Vendor: ATA      Model: TOSHIBA MK5056GS Rev: LJ00  
  Type:   Direct-Access          ANSI  SCSI revision: 05
```

El dispositivo sg permite a los usuarios enviar comandos a los dispositivos SCSI. También proporciona en la carpeta /proc/scsi/sg información sobre el subsistema SCSI:

```
# ls -1 /proc/scsi/sg  
allow_dio  
debug  
def_reserved_size  
device_hdr  
devices  
device_strs  
version
```


La versión del controlador sg se indica en /proc/scsi/sg/version:

```
# cat /proc/scsi/sg/version
30534      3.5.34 [20061027]
```

/proc/scsi/sg/device_strs muestra a su vez los dispositivos SCSI:

```
# cat device_strs
ATA      TOSHIBA MK5056GS   LJ00
HL-DT-ST DVD+-RW GS20N   A110
ATA      TOSHIBA MK5056GS   LJ00
```

/proc/scsi/sg/devices es una tabla con un resumen sobre los dispositivos:

```
# cat devices
0  0  0  0  0  1  31  0  1
1  0  0  0  5  1  1  0  1
2  0  0  0  0  1  31  0  1
```

La tabla está compuesta por 9 campos. Las cabeceras de estas columnas se encuentran en /proc/scsi/sg/device_hdr:

```
# cat device_hdr
host  chan  id  lun  type  opens  qdepth  busy  online
```

En resumen, tendremos lo siguiente:

host	channel	id	lun	type	opens	qdepth	busy	online
0	0	0	0	0	1	31	0	1
1	0	0	0	5	1	1	0	1
2	0	0	0	0	1	31	0	1

Significado de los campos:

- host permite indexar cada host.
- channel es el número de dispositivo.
- id es el identificador SCSI del dispositivo.
- lun es el número de unidad lógica del dispositivo.
- type es el tipo de dispositivo (0 = disco, 5 = CDROM...).
- opens es el número de aperturas por sd, sr o sr y sg.
- qdepth es el número de comandos que pueden estar pendientes entre el controlador del HBA (Host Bus Adapter) y el objetivo.
- busy indica el número de comandos en espera.
- Si online vale 0, el dispositivo está marcado como desconectado debido a algunos errores, y por lo tanto los comandos SCSI que se le envíen serán omitidos por el kernel de Linux. El estado online es 1.

ad. /proc/tty

/proc/tty/drivers contiene los dispositivos de tipo consola o puerto serie:

```
# cat /proc/tty/drivers
/dev/tty          /dev/tty          5          0 system:/dev/tty
/dev/console      /dev/console      5          1 system:console
/dev/ptmx         /dev/ptmx         5          2 system
/dev/vc/0         /dev/vc/0         4          0 system:vtmaster
rfcomm           /dev/rfcomm       216 0-255 serial
serial           /dev/ttyS         4 64-95 serial
pty_slave        /dev/pts          136 0-1048575 pty:slave
pty_master       /dev/ptm          128 0-1048575 pty:master
unknown         /dev/tty          4 1-63 console
```

/proc/tty/driver/serial contiene las estadísticas de uso y el estado de cada una de las líneas tty y serie.

```
# cat /proc/tty/driver/serial
serinfo:1.0 driver revision:
0: uart:unknown port:000003F8 irq:4
1: uart:unknown port:000002F8 irq:3
2: uart:unknown port:000003E8 irq:4
3: uart:unknown port:000002E8 irq:3
```

ae. /proc/uptime

Contiene el tiempo transcurrido en segundos desde el arranque del sistema y la cantidad de tiempo durante la cual el sistema ha estado inactivo:

```
# cat /proc/uptime
139984.98 279761.84
```

af. /proc/version

Se muestra la versión del kernel de Linux de forma detallada:

```
# cat /proc/version
Linux version 3.16.0-4-amd64 (debian-kernel@lists.debian.org)
(gcc version 4.8.4 (Debian 4.8.4-1) ) #1 SMP Debian 3.16.7-ckt25-2
(2016-04-08)
```

El kernel se compiló desde una distribución Debian con gcc version 4.8.4.

ag. /proc/vmallocinfo

Proporciona información acerca del espacio de direcciones virtuales del kernel:

```
# cat /proc/vmallocinfo
0xfffffc9000000000-0xfffffc900000081000 528384
alloc_large_system_hash+0x151/0x219 pages=128 vmalloc N0=128
0xfffffc900000081000-0xfffffc9000000c2000 266240
alloc_large_system_hash+0x151/0x219 pages=64 vmalloc N0=64
0xfffffc9000000c2000-0xfffffc9000000c4000 8192
alloc_targets+0x24/0xaa [dm_mod] pages=1 vmalloc N0=1
0xfffffc9000000c4000-0xfffffc9000000c8000 16384
acpi_os_map_memory+0x92/0x105 phys=1fff0000 ioremap

...

0xfffffffffa03a8000-0xfffffffffa03c7000 126976
module_alloc_update_bounds+0xb/0x55 pages=30 vmalloc N0=30
0xfffffffffa03c7000-0xfffffffffa03cd000 24576
module_alloc_update_bounds+0xb/0x55 pages=5 vmalloc N0=5
0xfffffffffa03f2000-0xfffffffffa03fc000 40960
module_alloc_update_bounds+0xb/0x55 pages=9 vmalloc N0=9
0xfffffffffa03fc000-0xfffffffffa042a000 188416
module_alloc_update_bounds+0xb/0x55 pages=45 vmalloc N0=45
```

ah. /proc/vmstat

Este archivo contiene diversas estadísticas acerca de la memoria virtual.

```
# cat /proc/vmstat
nr_free_pages 13624
nr_inactive_anon 53
nr_active_anon 17374
nr_inactive_file 49766
nr_active_file 27080
nr_unevictable 0

...

unevictable_pgs_mlockfreed 0
thp_fault_alloc 0
thp_fault_fallback 0
thp_collapse_alloc 0
thp_collapse_alloc_failed 0
thp_split 0
```

ai. /proc/zoneinfo

Este archivo muestra información sobre las zonas de memoria, lo que permite analizar el comportamiento de la memoria virtual:

```
# cat /proc/zoneinfo
Node 0, zone      DMA
  pages free      3978
    min          21
    low          26
    high         31
    scanned      0
    spanned     4080
    present     3922
```

```

nr_free_pages 3978
nr_inactive_anon 0
nr_active_anon 0
nr_inactive_file 0
nr_active_file 0
nr_unevictable 0
...
count: 173
high: 186
batch: 31
vm stats threshold: 12
all_unreclaimable: 0
start_pfn: 4096
inactive_ratio: 1

```

3. Carpetas de un proceso

En /proc, algunas carpetas contienen un nombre compuesto únicamente por cifras correspondientes a los PID de los procesos del sistema. Estos archivos contienen la actividad del proceso que representan. Por ejemplo, el directorio /proc/1 representa la actividad del proceso /sbin/init que posee el PID 1.

```

# ls /proc/1
attr          cwd           mem           oom_score_adj statm
autogroup     environ      mountinfo     pagemap       status
auxv          exe          mounts        personality    syscall
cgroup        fd           mountstats    root          task
clear_refs    fdinfo        net           sched          wchan
cmdline       io           ns            sessionid
comm          limits        numa_maps     smaps
coredump_filter loginuid      oom_adj       stack
cpuset        maps         oom_score     stat

```

a. /proc/1/cmdline

Contiene el comando ejecutado al arrancar el proceso:

```
# cat cmdline  
init [2]
```

b. /proc/1/cwd

Es un enlace simbólico que apunta a la carpeta de trabajo del proceso, en este caso, la raíz del sistema de archivos /:

```
# ls -l /proc/1/cwd  
lrwxrwxrwx 1 root root 0 abr 30 19:49 /proc/1/cwd -> /
```

c. /proc/1/envIRON

Este archivo muestra las variables de entorno (en mayúsculas) del proceso. Los valores se encuentran en minúsculas.

```
# cat /proc/1/envIRON  
SHLVL=1HOME=/init=/sbin/initTERM=linuxdrop_caps=BOOT_IMAGE=/boot/  
vmlinuz-3.16.0--amd64PATH=/sbin:/usr/sbin:/bin:/usr/binPWD=/  
rootmnt=/rootroot@debian:/proc/1#
```

d. /proc/1/exe

Es un enlace simbólico que apunta al ejecutable del proceso:

```
# ls -l exe  
lrwxrwxrwx 1 root root 0 abr 30 10:26 exe -> /sbin/init
```

e. /proc/1/fd

Es una carpeta en la que encontraremos todos los descriptores de archivo para un proceso, mostrando qué archivos o dispositivos utiliza:

```
# ls -l /proc/1fd  
lrwx----- 1 root root 64 ago 6 12:59 10 -> /run/initctl
```

f. /proc/1/maps

Este archivo contiene un mapa de la memoria con los ejecutables y librerías asociados a este proceso:

```
# cat maps
00400000-00409000 r-xp 00000000 fe:00 261803
/sbin/init
00608000-00609000 r--p 00008000 fe:00 261803
/sbin/init
00609000-0060a000 rw-p 00009000 fe:00 261803
/sbin/init
...
7f7cb4a45000-7f7cb4a46000 r--p 0001f000 fe:00 654105
/lib/x86_64-linux-gnu/ld-2.13.so
7f7cb4a46000-7f7cb4a47000 rw-p 00020000 fe:00 654105
/lib/x86_64-linux-gnu/ld-2.13.so
7f7cb4a47000-7f7cb4a48000 rw-p 00000000 00:00 0
7fffce853000-7fffce874000 rw-p 00000000 00:00 0
[stack]
7fffce963000-7fffce964000 r-xp 00000000 00:00 0
[vdso]
fffffffff600000-fffffffff601000 r-xp 00000000 00:00 0
[vsyscall]
```


g. /proc/1/smmaps

Este archivo también ofrece un mapa de la memoria:

```
# cat smaps
00400000-00409000 r-xp 00000000 fe:00 261803
/sbin/init
Size:                36 kB
Rss:                 32 kB
Pss:                 32 kB
Shared_Clean:        0 kB
Shared_Dirty:        0 kB
Private_Clean:       32 kB
Private_Dirty:       0 kB
Referenced:          32 kB
Anonymous:           0 kB
AnonHugePages:       0 kB
Swap:                0 kB
KernelPageSize:      4 kB
MMUPageSize:         4 kB
Locked:              0 kB
00608000-00609000 r--p 00008000 fe:00 261803
/sbin/init
Size:                4 kB
Rss:                 4 kB
Pss:                 4 kB
Shared_Clean:        0 kB
Shared_Dirty:        0 kB
Private_Clean:       0 kB
Private_Dirty:       4 kB
Referenced:          4 kB
Anonymous:           4 kB
AnonHugePages:       0 kB
Swap:                0 kB
```

```

KernelPageSize:      4 kB
MMUPageSize:         4 kB
Locked:              0 kB
00609000-0060a000 rw-p 00009000 fe:00 261803
/sbin/init
Size:                4 kB
Rss:                 4 kB
Pss:                 4 kB
Shared_Clean:        0 kB
Shared_Dirty:        0 kB
Private_Clean:       0 kB
Private_Dirty:       4 kB
Referenced:          4 kB
Anonymous:           4 kB
AnonHugePages:       0 kB
Swap:                0 kB
KernelPageSize:      4 kB
MMUPageSize:         4 kB
Locked:              0 kB

...

fffffffff600000-fffffffff601000 r-xp 00000000 00:00 0
[vsyscall]
Size:                4 kB
Rss:                 0 kB
Pss:                 0 kB
Shared_Clean:        0 kB
Shared_Dirty:        0 kB
Private_Clean:       0 kB
Private_Dirty:       0 kB
Referenced:          0 kB
Anonymous:           0 kB

```

```
AnonHugePages:      0 kB
Swap:               0 kB
KernelPageSize:     4 kB
MMUPageSize:        4 kB
Locked:             0 kB
```

h. /proc/1/mem

Este archivo contiene la memoria del proceso. No se puede leer por el usuario.

i. /proc/1/root

Es un enlace simbólico que apunta a la raíz del sistema de archivos:

```
# ls -l /proc/1/root
lrwxrwxrwx 1 root root 0 abr 30 14:22 /proc/1/root -> /
```

j. /proc/1/stat

Este archivo contiene el estado del proceso:

```
# cat /proc/1/stat
1 (systemd) S 0 1 1 0 -1 4202752 12711 385686 43 1172 12 58 454 207 20
0 1 0 1 181264384 1381 18446744073709551615 140180813500416
140180814685532 140732709280304 140732709277024 140180806151763
0 671173123 4096 1260 0 0 0 17 1 0 0 293 0 0 140180816783520
140180816904384 140180823134208 140732709281614 140732709281625
140732709281625 140732709281773 0
```

k. /proc/1/statm

Este archivo representa el estado de la memoria utilizada por el proceso:

```
# cat /proc/1/statm
```

```
8481 1129 688 62 0 458 0
```

Obtendremos una lista de siete valores:

- Tamaño total del proceso
- Tamaño del proceso residente en memoria
- Memoria compartida con otros procesos
- Tamaño del código cargado
- Tamaño de las librerías compartidas cargadas para el proceso
- Memoria utilizada por la pila del proceso
- Número de páginas modificadas por el programa

l. `/proc/1/status`

Este archivo contiene información acerca del proceso, como el id del proceso (PID), el identificador del proceso padre (PPID), el estado del proceso (S para sleeping, R para running...), identificadores de usuario (UID) y grupo (GID) reales y efectivos, el uso de memoria y las máscaras de bits que indican qué señales son interceptadas, ignoradas y bloqueadas, etc.

```
# cat /proc/1/status
```

```
Name:      init
```

```
State:      S (sleeping)
```

```
Tgid:      1
```

```
Pid:       1
```

```
Ppid:      0
```

```
TracerPid: 0
```

```
Uid:      0  0      0      0
```

```
Gid:      0  0      0      0
```

```
FDSize:    64
```

```
Groups:
```

```
VmPeak:     10652 kB
```

```
VmSize:     10648 kB
```

```
VmLck:      0 kB
```

```
VmPin:      0 kB
```

```
VmHWM:      836 kB
```

```
VmRSS:      836 kB
```

```
VmData:          188 kB
VmStk:           136 kB
VmExe:           36 kB
VmLib:           2044 kB
VmPTE:           44 kB
VmSwap:          0 kB
Threads:         1
SigQ: 0/3934
SigPnd:          0000000000000000
ShdPnd:          0000000000000000
SigBlk:          0000000000000000
SigIgn:          ffffffff57f0d8fc
SigCgt:          00000000280b2603
CapInh:          0000000000000000
CapPrm:          ffffffffffffffff
CapEff:          ffffffffffffffff
CapBnd:          ffffffffffffffff
Cpus_allowed:     3
Cpus_allowed_list: 0-1
Mems_allowed:     00000000,00000001
Mems_allowed_list: 0
voluntary_ctxt_switches: 31640
nonvoluntary_ctxt_switches: 63
```

4. Modificación de los parámetros del kernel

Hasta ahora hemos explorado una buena parte del pseudo-sistema de archivos procfs. También es posible cambiar la configuración del kernel de Linux escribiendo en algunos archivos de /proc/sys.

a. Modificación no permanente

Como usuario root, podemos utilizar un editor de texto, el comando echo o bien sysctl para efectuar la modificación. Dicho esto, en el próximo arranque, procfs se regenera y habremos perdido todas las modificaciones.

Sintaxis

```
sysctl <opción> <variable=valor>
```

La variable designa el archivo a editar en /proc y el valor se escribirá en este archivo.

Opciones útiles:

Opción		Descripción
corta	larga	
-w	-- write	Modifica los parámetros sysctl.
-a	--all	Muestra todos los valores disponibles.
-p	--load	Carga los parámetros sysctl desde el archivo por defecto /etc/sysctl.conf. Podemos especificar otro archivo: --load=mi_archivo.conf

Ejemplo

Impedir que su ordenador reciba pings sin crear reglas en el firewall.

Como recordaremos, el ping recibido es un mensaje ICMP ECHO REQUEST (tipo 8) y el ordenador responde con un mensaje ICMP ECHO REPLY (tipo 0).

El archivo `/proc/sys/net/icmp_echo_ignore_all` permite configurar el kernel para ignorar los mensajes ICMP ECHO entrantes.

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
0
```

El valor 0 lo activa y el valor 1 lo desactiva.

Reemplace 0 por 1 con echo:

```
# echo "1" > icmp_echo_ignore_all
```

O bien cambie 0 por 1 con sysctl:

```
# sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

Observe que la ruta no está compuesta por / sino por puntos.

Compruebe la modificación:

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
1
```

Reinicie el equipo, inicie una sesión y muestre el contenido del archivo `/proc/sys/net/ipv4/icmp_echo_ignore_all`:

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
0
```

O muestre el contenido con sysctl:

```
# sysctl net.ipv4.icmp_echo_ignore_all
net.ipv4.icmp_echo_ignore_all = 0
```

Hemos perdido el valor modificado de nuestra configuración.

b. Modificación permanente

Para realizar una modificación persistente, hay que introducir los parámetros en el archivo `/etc/sysctl.conf`.

```
# cat /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection
(reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
```



```

#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

```

Todas las líneas que comienzan con una almohadilla (#) son comentarios.

La modificación de este archivo puede efectuarse con un editor de texto o bien usando el comando echo.

Ejemplo

Impedir que su ordenador reciba pings sin crear reglas en el firewall.

Como en el anterior ejemplo, necesitamos cambiar 0 por 1 en el archivo /proc/sys/net/ipv4/icmp_echo_ignore_all con echo:

```
# echo "net.ipv4.icmp_echo_ignore_all = 1" >> /etc/sysctl.conf
```

Compruebe la modificación:

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all  
1
```

Reinicie el equipo, inicie una sesión y muestre el contenido del archivo /proc/sys/net/ipv4/icmp_echo_ignore_all:

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all  
1
```

O muestre el contenido con sysctl:

```
# sysctl net.ipv4.icmp_echo_ignore_all  
net.ipv4.icmp_echo_ignore_all = 1
```

De esta forma conservamos la modificación.

sysfs

1. Presentación

Introducido por el kernel de Linux 2.6, el pseudo-sistema de archivos sysfs es, como procfs, un sistema de archivos virtual montado en la carpeta /sys. No ocupa espacio en disco y su tamaño es de 0 KB:

```
# ls -ld /sys
dr-xr-xr-x 13 root root 0 abr 30 14:22 /sys
```

Ya que procfs se encuentra saturado con una gran cantidad de información no relacionada con los procesos, se diseñó sysfs con el fin de exportar desde el espacio del kernel (Anillo 0) al espacio del usuario (Anillo 3) información sobre los dispositivos y sus controladores.

2. Principales archivos de /sys

El primer nivel /sys contiene las siguientes carpetas:

```
# ls /sys
block    bus      class    dev      devices  firmware
fs       hypervisor  kernel  module   power
```

a. /sys/block

Contiene información acerca de los dispositivos de bloques:

```
# ls /sys/block
dm-0 dm-1 dm-2 dm-3 loop0 loop1 loop2 loop3 loop4 loop5
loop6 loop7      sda sr0
```

Encontraremos, por ejemplo, un archivo en cada carpeta /sys/block/<disco>/stat que muestra las estadísticas de E/S (I/O) del disco:

```
# cat /sys/block/dm-0/stat
24113      0    551642    249092    12764      0    263840
532832     0    88828     781920
```

/sys/block/<disco>/<partición>/stat muestra estadísticas de E/S de la partición. El formato del archivo es idéntico al del disco.

b. /sys/bus

Contiene carpetas que representan cómo están conectados los dispositivos a los diferentes buses:

```
# ls /sys/bus
ac97  acpi  cpu  event_source  hid  i2c  machinecheck  pci
pci_express  platform  pnp  scsi  serio  spi  usb  xen  xen-
backend
```

c. /sys/class

Contiene una lista de carpetas que muestran los dispositivos agrupados en clases:

```
# ls /sys/class/
ata_device      bdi          dma          graphics      input
net             printer      scsi_device  sound         vc
ata_link        block        dmi          hidraw        leds
pci_bus         regulator    scsi_disk    spi_master    vtconsole
ata_port        bluetooth    drm          hwmon         mem
power_supply    rfkill       scsi_generic thermal        backlight
bsg             firmware     i2c-adapter  misc          ppdev
rtc             scsi_host    tty
```

d. /sys/dev

Proporciona dos carpetas:

```
# ls /sys/dev
block  char
```

Una está dedicada a los dispositivos de bloque (block) y la otra a los dispositivos de caracteres (char). En esta última, los nombres de los enlaces simbólicos son los números mayores y menores de los dispositivos:

```
# ls -l /sys/dev/block
total 0
lrwxrwxrwx 1 root root 0 abr 3 10:26 11:0 -
.././devices/pci0000:00/0000:00:01.1/host2/target2:0:0/2:0:0:0/
block/sr0
lrwxrwxrwx 1 root root 0 abr 8 00:36 254:0 ->
.././devices/virtual/block/dm-0
lrwxrwxrwx 1 root root 0 abr 8 00:36 254:1 ->
.././devices/virtual/block/dm-1
lrwxrwxrwx 1 root root 0 abr 3 10:26 254:2 ->
.././devices/virtual/block/dm-2
lrwxrwxrwx 1 root root 0 abr 3 10:26 254:3 ->
.././devices/virtual/block/dm-3
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:0 ->
.././devices/virtual/block/loop0
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:1 ->
.././devices/virtual/block/loop1
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:2 ->
.././devices/virtual/block/loop2
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:3 ->
.././devices/virtual/block/loop3
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:4 ->
.././devices/virtual/block/loop4
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:5 ->
.././devices/virtual/block/loop5
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:6 ->
.././devices/virtual/block/loop6
lrwxrwxrwx 1 root root 0 abr 8 00:36 7:7 ->
.././devices/virtual/block/loop7
lrwxrwxrwx 1 root root 0 abr 3 10:26 8:0 ->
```

```
../../devices/pci0000:00/0000:00:0d.0/host0/target0:0:0/0:0:0:0/  
block/sda  
lrwxrwxrwx 1 root root 0 abr  3 10:26 8:1 ->  
../../devices/pci0000:00/0000:00:0d.0/host0/target0:0:0/0:0:0:0/  
block/sda/sda1
```

Los enlaces simbólicos apuntan a la carpeta /sys/devices.

e. /sys/devices

Esta carpeta contiene subcarpetas que representan la capa física:

```
# ls /sys/devices  
breakpoint  LNXSYSTM:00  pci0000:00  platform  pnp0  software  
system  tracepoint  virtual
```

f. /sys/firmware

El BIOS o firmware proporciona una memoria que el kernel de Linux lee. Esta memoria se muestra en el espacio del usuario a través de /proc/iomem.

/sys/firmware contiene las carpetas:

```
# ls /sys/firmware/memmap  
0  1  2  3  4  5
```

El número de carpetas depende de la cantidad de memoria.

Cada carpeta contiene tres archivos:

- start que contiene la dirección de inicio;
- end que contiene la dirección final;
- type que contiene el tipo de entrada. Los tipos pueden ser System RAM, ACPI tablas, ACPI Non-volatile Storage y reserved.

Ejemplo

El objetivo es mostrar la información de memmap de manera más legible para un ser humano empleando un script de shell.

Debe estar conectados como root. Si es necesario, vaya al directorio /root. Compruebe con el comando pwd.

Abra un editor de texto como vi o nano para crear el archivo memmap.sh.

A continuación, escriba las siguientes líneas:

```
#!/bin/bash
cd /sys/firmware/memmap

for dir in * ; do
    start=$(cat $dir/start)
    end=$(cat $dir/end)
    type=$(cat $dir/type)
    printf "%016x-%016x (%s)\n" $start $[ $end +1] "$type"
done
```

Guarde el archivo y salga del editor de texto.

Asigne permisos de ejecución al archivo memmap.sh:

```
# chmod +x memmap.sh
```

Ejecute el script:

```
# ./memmap.sh
0000000000000000-000000000009fc00 (System RAM)
000000000009fc00-00000000000a0000 (reserved)
00000000000f0000-0000000000100000 (reserved)
0000000000100000-000000001ffff000 (System RAM)
000000001ffff000-0000000020000000 (ACPI Tables)
00000000fffc0000-0000000100000000 (reserved)
```

g. /sys/fs

Esta carpeta contiene la información sobre los sistemas de archivos montados como ext4 y también sobre la funcionalidad del kernel cgroups (control groups) que permite limitar, contar y aislar el uso de recursos como el procesador, la memoria, etc.

```
# ls /sys/fs/*  
  
/sys/fs/cgroup:  
  
/sys/fs/ext4:  
dm-0  dm-2  dm-3  features
```

h. /sys/hypervisor

Esta carpeta contiene información acerca del hypervisor Xen cuando está instalado.

i. /sys/kernel

Este árbol contiene información sobre el kernel:

```
# ls /sys/kernel/*  
  
/sys/kernel/fscaps          /sys/kernel/kexec_crash_size  
/sys/kernel/notes          /sys/kernel/uevent_helper  
/sys/kernel/vmcoreinfo  
/sys/kernel/kexec_crash_loaded  /sys/kernel/kexec_loaded  
/sys/kernel/profiling/sys/kernel/uevent_seqnum  
  
/sys/kernel/debug:  
  
/sys/kernel/fscache:  
  
/sys/kernel/mm:  
hugepages  ksm          transparent_hugepage  
  
/sys/kernel/security:  
  
/sys/kernel/debug
```


Esta carpeta la utiliza el sistema de archivos debugfs, y permite depurar el código del kernel de Linux. No debemos confundir este sistema de archivos debugfs con la herramienta debugfs.

La opción debugfs debe activarse en la configuración del kernel durante su compilación. Por lo general, las distribuciones recientes incluyen esta opción.

Ubuntu Server 12.04 LTS monta por defecto el sistema de archivos debugfs.

```
# mount | grep debugfs
none on /sys/kernel/debug type debugfs (rw)
```

Para Debian y CentOS, debemos montarlo manualmente:

```
# mount -t debugfs nodev /sys/kernel/debug
```

Si lo queremos desmontar:

```
# umount nodev
```

Para comprobar el montaje:

```
# mount | grep debugfs
none on /sys/kernel/debug type debugfs (rw)
```

Por último, si desea que el montaje esté activo durante el arranque del sistema, basta con añadir la siguiente línea al archivo /etc/fstab:

```
debugfs /sys/kernel/debug debugfs defaults 0 0
```

Una vez montado debugfs, dispone de este árbol:

```
# ls debug/*
debug/gpio  debug/sched_features  debug/suspend_stats  debug/
wakeup_sources

debug/acpi:

debug/bdi:
11:0  254:0  254:1  254:2  254:3  7:0  7:1  7:2  7:3  7:4  7:5
      7:6  7:7  8:0  default
```

debug/bluetooth:

l2cap rfcomm rfcomm_dlc sco

debug/dri:

0

debug/extfrag:

extfrag_index unusable_index

debug/hid:

0003:80EE:0021.0001

debug/kprobes:

enabled list

debug/mce:

fake_panic severities-coverage

debug/regmap:

debug/regulator:

dummy supply_map

debug/tracing:

available_events buffer_total_size_kb free_buffer
printk_formats set_event trace_marker tracing_cpumask
tracing_thresh
available_tracers current_tracer options README
trace trace_options tracing_enabled
buffer_size_kb events per_cpu
saved_cmdlines trace_clock trace_pipe tracing_on

```
debug/usb:
devices

debug/x86:
pat_memtype_list
```

De esta forma, dispone de dos opciones:

- Incluir en su código fuente (escrito en lenguaje C) las APIs de debugfs. En este caso, es necesario incluir el archivo de cabecera debugfs.h ubicado en la carpeta /usr/src/linux-headers-\$(uname -r)-common/include/linux.
- Utilizar la herramienta ftrace. Asegúrese de que las opciones de configuración Debugfs, FUNCTION_TRACER, FUNCTION_GRAPH_TRACER, DYNAMIC_FTRACE y STACK_TRACER están activadas durante la compilación del kernel de Linux.

j. /sys/module

Esta carpeta contiene todos los módulos del kernel (Loadable Kernel Modules) del sistema operativo:

```
# ls /sys/module
8250      crc_t10dif  libahci    ppdev      snd_timer
ac        dm_mod      libata     printk     soundcore
ac97_bus  dns_resolver lockd      processor  spurious
acpi      drm         loop       psmouse    sr_mod
acpiphp   e1000       lp         pstore     sunrpc
ahci      ehci_hcd    mbcache    rcutree    tcp_cubic
apparmor  evdev       mousedev   rfcomm     thermal_sys
ata_generic ext4        netpoll    rfkill     uinput
ata_piix  fscache     nfs        scsi_mod   usb_common
auth_rpcgss hid         nfs_acl    sd_mod     sbcore
battery   i2c_core    nfsd       serio_raw  usbhid
binfmt_misc i2c_piix4  ohci_hcd   sg         vboxguest
block     i8042       parport    snd         vboxvideo
bluetooth intel_idle  parport_pc snd_ac97_codec vt
bnep      ipv6        pcie_aspm  snd_intel8x0 xz_dec

button    jbd2        pciehp     snd_page_alloc
```

cdrom	joydev	pci_hotplug	snd_pcm
cpuidle	kernel	pcspkr	snd_seq
crc16	keyboard	power_supply	snd_seq_device

k. /sys/power

La carpeta /sys/power contiene todos los archivos que proporcionan una interfaz unificada para el subsistema de gestión de la alimentación (power management):

```
# ls /sys/power
disk image_size pm_async pm_test reserved_size resume state
wakeup_count
```

/sys/power/state

Este archivo controla el estado del nivel de ahorro de energía utilizado por el equipo. Los posibles valores son:

- standby (Power-On Suspend). Corresponde a poner el equipo en estado de espera.
- mem (Suspend-to-RAM). Permite la suspensión del sistema en memoria.
- disk (Suspend-to-Disk). Permite apagar completamente el equipo (es decir, de hibernar el equipo) después de haber almacenado su estado en la partición de intercambio (swap). Se recomienda por ello que esta partición sea al menos igual en tamaño a la RAM.

/sys/power/disk

Este archivo contiene la información para el mecanismo de hibernación.

/sys/power/image_size

Este archivo controla el tamaño de la imagen creada durante la hibernación.

3. Herramienta systool

El comando systool muestra información de los dispositivos del sistema por bus, categoría y topología. Esta herramienta debe instalarse empleando el paquete sysfsutils.

a. Instalación

Debian y Ubuntu

La instalación se realiza de esta manera:

```
# apt-get -y install sysfsutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libsysfs2
Se instalarán los siguientes paquetes NUEVOS:
  libsysfs2 sysfsutils
0 actualizados, 2 se instalarán, 0 para eliminar y 4 no actualizados.
Necesito descargar 32,0 kB de archivos.
Se utilizarán 182 kB de espacio de disco adicional después de esta
operación.
Des: 1 http://es.archive.ubuntu.com/ubuntu/ trusty/main libsysfs2 amd64
2.1.0+repack-3ubuntu1 [19,3 kB]
Des: 2 http://es.archive.ubuntu.com/ubuntu/ trusty/universe sysfsutils
amd64 2.1.0+repack-3ubuntu1 [12,7 kB]
Descargados 32,0 kB en 0seg. (40,5 kB/s)
Seleccionando el paquete libsysfs2:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 199461 ficheros o directorios instalados
actualmente.)
Preparing to unpack .../libsysfs2_2.1.0+repack-3ubuntu1_amd64.deb ...
Unpacking libsysfs2:amd64 (2.1.0+repack-3ubuntu1) ...
Seleccionando el paquete sysfsutils previamente no seleccionado.
Unpacking sysfsutils (2.1.0+repack-3ubuntu1) ...

Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Configurando libsysfs2:amd64 (2.1.0+repack-3ubuntu1) ...
Configurando sysfsutils (2.1.0+repack-3ubuntu1) ...
* Setting sysfs variables... [ OK ]
```

CentOS

La instalación de systool se realiza así:

```
# yum -y install sysfsutils
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
* base: ftp.ciril.fr
* extras: mirror.in2p3.fr
* updates: ftp.ciril.fr
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package sysfsutils.x86_64 0:2.1.0-7.el6 will be installed
--> Processing Dependency: libsysfs.so.2()(64bit) for package:
sysfsutils-2.1.0-7.el6.x86_64
--> Running transaction check
---> Package libsysfs.x86_64 0:2.1.0-7.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package            Arch      Version      Repository    Size
=====
Installing:
 sysfsutils        x86_64    2.1.0-7.el6  base          38 k
Installing for dependencies:
 libsysfs          x86_64    2.1.0-7.el6  base          44 k

Transaction Summary
=====
Install            2 Package(s)

Total download size: 82 k
Installed size: 256 k
```

Downloading Packages:

(1/2): libsysfs-2.1.0-7.el6.x86_64.rpm | 44 kB 00:00

(2/2): sysfsutils-2.1.0-7.el6.x86_64.rpm | 38 kB 00:00

Total 231 kB/s | 82 kB 00:00

Running rpm_check_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : libsysfs-2.1.0-7.el6.x86_64 1/2

Installing : sysfsutils-2.1.0-7.el6.x86_64 2/2

Verifying : sysfsutils-2.1.0-7.el6.x86_64 1/2

Verifying : libsysfs-2.1.0-7.el6.x86_64 2/2

Installed:

sysfsutils.x86_64 0:2.1.0-7.el6

Dependency Installed:

libsysfs.x86_64 0:2.1.0-7.el6

Complete!

b. Uso de systool

La ejecución de systool sin argumentos muestra todos los buses, las clases y los dispositivos raíz que están disponibles:

```
# systool
Supported sysfs buses:
  ac97
  acpi
  cpu
  ...
  vtconsole
  ...
  usb_common
  usbcore
  usbhid
  vboxguest
  vboxvideo
  vt
  xz_dec
```

Sintaxis

```
systool <opción>
```

Las opciones son:

Opción corta	Descripción
-a	Muestra los atributos del recurso solicitado.
-b	Muestra la información para un bus específico.
-c	Muestra la información para una clase de dispositivo específica.
-d	Muestra solo los dispositivos.

-h	Muestra la ayuda del comando.
-m	Muestra la información para un módulo específico.
-p	Muestra la ruta absoluta de un recurso en sysfs.
-v	Muestra todos los atributos con sus valores.
-A	Muestra los atributos del recurso solicitado.
-D	Muestra solo los controladores de dispositivo.
-P	Muestra el dispositivo padre.

Mostrar el bus SCSI

```
# systool -b scsi
Bus = "scsi"

Device = "0:0:0:0"
Device = "2:0:0:0"
Device = "host0"
Device = "host1"
Device = "host2"

Device = "target0:0:0"
Device = "target2:0:0"
```

Mostrar el bus y la ruta absoluta en sysfs

```
# systool -b scsi -p
Bus = "scsi"

Device = "0:0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:0d.0/host0/target0:0:0/0:0:0:0"

Device = "2:0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:01.1/host2/target2:0:0/2:0:0:0"

Device = "host0"
Device path = "/sys/devices/pci0000:00/0000:00:0d.0/host0"

Device = "host1"
Device path = "/sys/devices/pci0000:00/0000:00:01.1/host1"

Device = "host2"
Device path = "/sys/devices/pci0000:00/0000:00:01.1/host2"

Device = "target0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:0d.0/host0/target0:0:0"

Device = "target2:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:01.1/host2/target2:0:0"
```

Mostrar el bus SCSI y mostrar los atributos con sus valores

```
# systool -b scsi -v
Bus = "scsi"

Device = "0:0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:0d.0/host0/target0:0:0/0:0:0:0"

delete                = <store method only>
device_blocked        = "0"
evt_media_change      = "0"
iocounterbits         = "32"
iodone_cnt             = "0xf65e"
ioerr_cnt             = "0x3"
iorequest_cnt         = "0xf67e"
modalias              = "scsi:t-0x00"
model                 = "VBOX HARDDISK  "
queue_depth           = "31"
queue_ramp_up_period= "120000"
queue_type            = "simple"
rescan                = <store method only>
rev                   = "1.0 "
scsi_level             = "6"
state                 = "running"
timeout               = "30"
type                  = "0"
uevent                = "DEVTYPE=scsi_device"
DRIVER=sd
MODALIAS=scsi:t-0x00"
vendor                = "ATA  "
```

```

Device = "2:0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:01.1/host2/target2:0:0/2:0:0:0"
    delete                = <store method only>
    device_blocked         = "0"
    evt_media_change       = "0"
    iocounterbits          = "32"
    iodone_cnt              = "0x17"
    ioerr_cnt              = "0x2"
    iorequest_cnt          = "0x27"
    modalias               = "scsi:t-0x05"
    model                  = "CD-ROM          "
    queue_depth            = "1"
    queue_type              = "none"
    rescan                  = <store method only>
    rev                    = "1.0 "
    scsi_level              = "6"
    state                  = "running"
    timeout                = "30"
    type                   = "5"
    uevent                 = "DEVTYPE=scsi_device"
DRIVER=sr
MODALIAS=scsi:t-0x05"
    vendor                 = "VBOX      "

Device = "host0"
Device path = "/sys/devices/pci0000:00/0000:00:0d.0/host0"
    uevent                 = "DEVTYPE=scsi_host"

Device = "host1"
Device path = "/sys/devices/pci0000:00/0000:00:01.1/host1"
    uevent                 = "DEVTYPE=scsi_host"

```

```
Device = "host2"
Device path = "/sys/devices/pci0000:00/0000:00:01.1/host2"

uevent          = "DEVTYPE=scsi_host"

Device = "target0:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:0d.0/host0/target0:0:0"
uevent          = "DEVTYPE=scsi_target"

Device = "target2:0:0"
Device path =
"/sys/devices/pci0000:00/0000:00:01.1/host2/target2:0:0"
uevent          = "DEVTYPE=scsi_target"
```

4. Modificación de los parámetros del kernel

Si planea modificar un archivo en /sys, no existe ninguna herramienta igual a sysctl. Sin embargo, puede usar el comando echo para un cambio no permanente. Si queremos mantener los parámetros de forma permanente entonces escribimos las directivas en /etc/rc.local.

Ejemplo

El «kernel polling» es una alternativa al tratamiento básico de interrupciones. El kernel comprobará periódicamente un dispositivo sin ser interrumpido.

No se encuentra activado por defecto en algunas distribuciones. Para consultar el parámetro global, escriba:

```
# cat /sys/module/block/parameters/events_dfl_poll_msecs
-1
```

O para consultar el parámetro del lector de CDROM:

```
# cat /sys/block/sr0/events_poll_msecs
-1
```

Si el valor es 0 o -1 entonces la característica está desactivada. El valor razonable (en milisegundos) debe estar comprendido entre 2000 y 5000. Consultas demasiado frecuentes desperdician los recursos de la CPU.

Desactivar el "kernel polling" del lector de CDROM (/dev/SR0) puede provocar que la detección de medios no funcione y la desactivación del botón de expulsión.

Configure el valor 2000 en el archivo /sys/block/sr0/events_poll_msecs:

```
# echo 2000 > /sys/block/sr0/events_poll_msecs
```

Para conservar este parámetro de forma definitiva, agregue con un editor de texto la línea anterior al archivo /etc/rc.local antes de efectuar un exit 0:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

echo 2000 > /sys/block/sr0/events_poll_msecs
exit 0
```

Ejercicio

El objetivo es modificar de manera permanente la gestión del swap.

1. Inicie una sesión como usuario root.
2. ¿Cuál es el parámetro del kernel de Linux que controla la cantidad de datos que debe conservar la RAM o ponerse en el swap? Muestre su valor.
3. Cambie el valor de este parámetro a 10. Asegúrese de que este cambio sea permanente. Reinicie el sistema y verifique.

Tipos de problemas de hardware

Los problemas de hardware pueden clasificarse en dos categorías:

- error de hardware
- hardware no detectado por el sistema

1. Problemas de hardware

Es cierto que los fallos de hardware no conciernen de forma directa al sistema operativo Linux. Sin embargo, si no tenemos un hardware operativo, entonces el sistema no funcionará. El uno depende del otro.

a. Hardware del equipo que no funciona

Un componente fuera de servicio suele ser el caso más simple. Si su fuente de alimentación está muerta, la máquina no arrancará. Es evidente.

El problema puede ocurrir durante el encendido. La BIOS realiza un POST (Power-On Self Test) para verificar el correcto funcionamiento del equipo:

- la inicialización del procesador (CPU - Central Processing Unit);
- la estabilidad de la alimentación;
- la integridad del código de la BIOS;
- la integridad de la placa base;
- la inicialización de entradas y salidas (E/S o I/O);
- la visualización en pantalla de un posible mensaje indicando pulsar una tecla como [Supr] o [F2] para acceder al setup de la BIOS. Una señal sonora permite identificar a veces la etapa del inicio de E/S que ha fallado;
- la identificación del programa a ejecutar;
- la inicialización del equipo puede variar en función de su configuración...

El sistema arranca «a priori» correctamente y, por ejemplo, podemos tener una tarjeta gráfica defectuosa que provoca una resolución de pantalla más baja, una tarjeta de audio que no emite sonido, un error de paridad en la memoria, etc.

En algunos casos, el sistema operativo puede mostrar por pantalla mensajes de error o escribir en un registro. O por el contrario, se congela sin mensajes en pantalla. ¿Habría tenido tiempo de escribir en un registro? Esto no es seguro...

b. Funcionamiento errático del hardware

Un componente de hardware provoca errores solo de vez en cuando. No contamos con suficiente información para relacionar todos estos fallos por separado. ¿Cómo encontrar la o las causas?

Además, es posible que veamos varios síntomas que en apariencia no afectan al equipo, e incluso podrían proceder de varias fuentes.

c. Problemas de firmware y controlador

Un problema de controlador de dispositivo por lo general se manifiesta de manera similar a un fallo de hardware.

¿Tenemos el controlador correcto instalado? Puede estar proporcionado por la distribución o haber sido añadido por nosotros. Su código fuente es genérico o bien está desarrollado por el fabricante.

¿Tenemos la versión correcta del controlador? La versión utilizada presenta un error que puede ser corregido con una versión más reciente. O por el contrario, el controlador es demasiado reciente. Si bien se considera estable aunque no está demostrado, e incluso probado. En este caso, será necesario sustituirlo por la versión anterior.

d. Otras consideraciones

Un ordenador puede también tener una BIOS bloqueada que le impide sacar el máximo provecho de los componentes de hardware o de sus funcionalidades.

Puede que usemos componentes de hardware que no están diseñados para operar juntos.

2. Hardware no detectado por el sistema

Un hardware de su servidor o de su estación de trabajo Linux no funciona a pesar de estar bien conectado. La primera etapa consiste en comprobar si es compatible con Linux. Para ello, podemos consultar en sitios web la HCL (Hardware Compatibility List). He aquí algunas direcciones de sitios que pueden ayudarnos en nuestra búsqueda:

Sitio	Descripción
http://www.linuxquestions.org/hcl/	Foro generalista (no oficial).
https://wiki.debian.org/Hardware	Sitio oficial Debian
http://www.ubuntu.com/certification/	Sitio oficial Ubuntu
https://hardware.redhat.com/	Sitio HCL oficial de Red Hat.

Los proveedores producen hardware diseñado con solo algunos sistemas operativos en mente, puede que nunca contemos con controladores oficiales para Linux. En este caso, podemos intentar probar con controladores alternativos. Por ejemplo: las impresoras Lexmark y PCL.

Analizar el hardware

Disponemos de una multitud de herramientas nativas o a instalar para:

- Conocer los componentes del equipo.
- Verificar si el sistema detecta correctamente el hardware. Si es el caso, no queda más que encontrar el controlador adecuado para el dispositivo que probablemente sea un módulo del kernel para cargar.

1. Enumerar el hardware

El comando `lshw` muestra el hardware. Obtiene su información de varias fuentes:

- El archivo `pci.ids`. Es un registro público de todos los ID (identificaciones) conocidos que son utilizados por los dispositivos PCI. Se ubica en la carpeta `/usr/share/hwdata` o `/usr/share/misc` según la distribución de Linux.
- Los archivos `/proc/bus/pci`, `/proc/ide`, `/proc/scsi` `/dev/sg*`, `/dev/CPU`, `/proc/device-tree`, `/proc/bus/usb`, `/sys/*`.

Cabe señalar que puede tener información más o menos correcta sobre Firewire (IEEE1394) y las interfaces SCSI virtuales en un bus IDE.

a. Instalación de lshw

Este comando no se instala de forma predeterminada en todas las distribuciones de Linux. Sin embargo, el proyecto lshw puede descargarse del sitio web <http://ezix.org/project/wiki/HardwareLiSter>.

Debian

lshw se encuentra en los repositorios. La instalación se realiza como sigue:

```
# apt-get install lshw
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  lshw
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 234,0 kB de archivos.
Se utilizarán 814 kB de espacio de disco adicional después de esta
operación.
Des: 1 http://ftp.es.debian.org/debian/ jessie/main lshw
amd64 02.17-1.1 [234 kB]
Descargados 234 kB en 0s (864 kB/s)
Seleccionando el paquete lshw previamente no seleccionado.
(Leyendo la base de datos ... 247269 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../lshw_02.17-1.1_amd64.deb ...
Desempaquetando lshw (02.17-1.1) ...
Procesando disparadores para man-db (2.7.0.2-5) ...Processing triggers
for man-db (2.6.7.1-1ubuntu1) ...
Configurando lshw (02.17-1.1) ...
```

Ubuntu Server

El comando se instala por defecto. No necesitamos instalarlo.

CentOS

CentOS no integra lshw en sus repositorios. Es necesario encontrar el paquete en Internet utilizando un motor de búsqueda de paquetes RPM.

Como nos hemos referido en el capítulo Arquitectura del sistema GNU/Linux - Distribuciones, CentOS 7 es una distribución basada en la compilación del código fuente de Red Hat Enterprise Linux 7.

La descarga de lshw se realiza desde un navegador web con el protocolo HTTP o FTP o bien desde un terminal con el comando wget.

Sintaxis

```
wget <URL>
```

La URL (Uniform Resource Locator) es HTTP, HTTPS y FTP.

El comando rpm (Red Hat Package Manager) instala el paquete.

Sintaxis

```
rpm <opciones> <nombre_del_paquete_RPM>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-i	--install	Instala el paquete especificado.
-v	(ninguna)	Muestra información detallada (modo «verboso»).
-h	--hash	Muestra una barra de progreso durante la instalación del paquete.

Instalación del paquete lshw 64 bits para CentOS 6

A través de un motor de búsqueda como pbone.net, encontramos el paquete RPM: lshw-B.02.17-2.el7.x86_64.rpm.

La versión B.02.17-2 64 bits (x86_64) de lshw soporta la versión de CentOS 7 o Red Hat 7.

Abra un terminal y descargue el paquete RPM:

```
# wget ftp://ftp.icm.edu.pl/vol/rzm5/linux-slc/centos/7.0.1406/  
cr/x86_64/Packages/lshw-B.02.17-2.el7.x86_64.rpm
```

Procedemos a la instalación del paquete RPM:

```
# rpm -ivh lshw-B.02.17-2.el7.x86_64.rpm
Preparando... ##### [100%]
Actualizando / instalando...
  1:lshw-B.02.17-2.el7 ##### [100%]
```

Probamos el comando:

```
# lshw -version
B.02.17
```

La versión es, en el momento de la redacción de este libro, B.02.17.

b. Utilización de lshw

Sin opciones, el comando lshw recoge el conjunto del hardware detectado:

```
# lshw
debian
  description: Computer
  width: 64 bits
  capabilities: smbios-2.5 vsyscall32
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: *-memory
  description: System memory
  physical id: 0
  size: 504MiB
*-cpu
  product: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0

  width: 64 bits
```

```
capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8
apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht
syscall nx rdtscp x86-64 constant_tsc rep_good nopl xtopology
nonstop_tsc pn1 pclmulqdq ssse3 cx16 sse4_1 sse4_2 popcnt aes xsave
avx hypervisor lahf_lm
```

*** Salida truncada ***

Permite también una visualización por clase.

Sintaxis

```
lshw <opción> <nombre_de_la_clase>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-c o -C	-class	Muestra la clase del hardware solicitado.
(ninguna)	-businfo	Muestra una lista de dispositivos con información sobre los buses.
(ninguna)	-short	Muestra el árbol de dispositivos mostrando las rutas del hardware.

Para conocer la lista de clases, escribimos en un terminal:

<https://dogramcode.com/libros-sistemas>

``` # lshw -businfo ```

Bus info	Device	Class	Description
=====			
		system	VirtualBox
		bus	VirtualBox
		memory	128KiB BIOS
		memory	496MiB System memory
cpu@0		processor	Intel(R) Core(TM)
i7-2600 CPU	@ 3.40GHz		
pci@0000:00:00.0		bridge	440FX - 82441FX PMC
	[Natoma]		
pci@0000:00:01.0		bridge	82371SB PIIX3 ISA
	[Natoma/Triton II]		
pci@0000:00:01.1		storage	82371AB/EB/MB PIIX4
	IDE		
pci@0000:00:02.0		display	VirtualBox Graphics
	Adapter		
pci@0000:00:03.0	eth0	network	82540EM Gigabit
	Ethernet Controller		
pci@0000:00:04.0		generic	VirtualBox Guest
	Service		
pci@0000:00:05.0		multimedia	82801AA AC'97 Audio
	Controller		
pci@0000:00:06.0		bus	KeyLargo/Intrepid USB
pci@0000:00:07.0		bridge	82371AB/EB/MB PIIX4
	ACPI		
pci@0000:00:0b.0		bus	82801FB/FBM/FR/FW/FRW
	(ICH6 Family) USB2 EHCI Controller		
pci@0000:00:0d.0		storage	82801HM/HEM
	(ICH8M/ICH8M-E) SATA Controller [AHCI mode]		
	scsi0	storage	
scsi@0:0.0.0	/dev/sda	disk	53GB VBOX HARDDISK
scsi@0:0.0.0,1		/dev/sda1	volume 49GiB Linux LVM

Physical Volume partition

	scsi2	storage	
scsi@2:0.0.0	/dev/cdrom1	disk	DVD reader

O bien:

lshw -short

H/W path	Device	Class	Description
=====			
		system	VirtualBox ()
/0		bus	VirtualBox
/0/0		memory	128KiB BIOS
/0/1		memory	496MiB System memory
/0/2		processor	Intel(R) Core(TM)
i7-2600 CPU @ 3.40GHz			
/0/100		bridge	440FX - 82441FX PMC
[Natoma]			
/0/100/1		bridge	82371SB PIIX3 ISA
[Natoma/Triton II]			
/0/100/1.1		storage	82371AB/EB/MB PIIX4
IDE			
/0/100/2		display	VirtualBox Graphics
Adapter			
/0/100/3	eth0	network	82540EM Gigabit
Ethernet Controller			
/0/100/4		generic	VirtualBox Guest
Service			
/0/100/5		multimedia	82801AA AC'97 Audio
Controller			
/0/100/6		bus	KeyLargo/Intrepid USB
/0/100/7		bridge	82371AB/EB/MB PIIX4
ACPI			
/0/100/b		bus	82801FB/FBM/FR/FW/FRW

(ICH6 Family) USB2 EHCI Controller

/0/100/d storage 82801HM/HEM

(ICH8M/ICH8M-E) SATA Controller [AHCI mode]

/0/3 scsi0 storage

/0/3/0.0.0 /dev/sda disk 53GB VBOX HARDDISK

/0/3/0.0.0/1 /dev/sda1 volume 49GiB Linux LVM

Physical Volume partition

/0/4 scsi2 storage

/0/4/0.0.0 /dev/cdrom1 disk DVD reader

La tabla de clases disponibles:

Clase	Información acerca de
address	Los rangos de memoria (extensión ROM, memoria de vídeo...).
bridge	Los convertidores de bus internos (PCI-to-PCI bridge, AGP bridge, controlador PCMCIA, host bridge...).
bus	Los buses (USB, SCSI, Firewire...).
communication	Los dispositivos de comunicación (modem, puertos serie...).
disk	Los dispositivos de almacenamiento (CD-ROM, DVD-RW...).
display	Los dispositivos de visualización (EGA / VGA, UGA...).
generic	Los dispositivos genéricos.
input	Los dispositivos de entrada (teclado, ratón, joystick...).
memory	La memoria.
multimedia	Los dispositivos de audio y vídeo (tarjeta de sonido, tarjetas de TV, tarjeta de captura de vídeo...).
network	Las interfaces de comunicación (Ethernet, FDDI, inalámbrico...).

power	La gestión de la energía (batería, SAI...).
printer	Los dispositivos de impresión.
processor	El o los procesadores.
storage	Los controladores de disco (controlador SCSI, controlador IDE...).
system	El equipo.
tape	Los dispositivos de almacenamiento (DAT, DDS...).
volume	Los volúmenes de discos (sistemas de archivo, swap...).

2. Sistema

La clase system de lshw recopilará la información del equipo en el que estamos trabajando:

```
# lshw -c system
debsrv
  descripción: Ordenador portátil
  producto: Precision M6500 ()
  fabricante: Dell Inc.
  número de serie: C42FF02
  bits: 64 bits
  capacidades: smbios-2.6 dmi-2.6 vsyscall32
  configuración: boot=normal chasis=portátil uuid=44454C4C-
3600-1035-8046-C3C04F375031
```

El equipo analizado es un portátil Dell Precision M6500.

3. CPU

El archivo /proc/cpuinfo contiene información sobre el o los procesadores del equipo. También contamos con el comando lscpu:

```
# lscpu
```

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Thread(s) per core:    1
Core(s) per socket:    2
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 42
Model name:            Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
Stepping:              7
CPU MHz:               3392.298
BogoMIPS:              6784.59
Hypervisor vendor:     KVM
Virtualization type:    full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              8192K
NUMA node0 CPU(s):     0,1
```

El procesador es un x86_64 I7 con un hilo de memoria por núcleo (core).

Con la clase processor de lshw obtendremos también algunos datos:

```
# lshw -c processor
*-cpu
```

```

product: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
vendor: Intel Corp.
physical id: 1
bus info: cpu@0
width: 64 bits
capabilities: fpu fpu_exception wp vme de pse tsc msr pae
mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse
sse2 ht syscall nx rdtscp x86-64 constant_tsc rep_good nopl
xtopology nonstop_tsc pni pclmulqdq ssse3 cx16 sse4_1 sse4_2 popcnt
aes xsave avx hypervisor lahf_lm

```

4. Tabla DMI

DMI (Desktop Management Interface) gestiona y supervisa los cambios de componentes en un equipo. El comando `dmidecode` permite leer esta información de manera comprensible para un ser humano.

Sintaxis

```
dmidecode <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-d	--dev-mem	Lee la información desde un archivo. Por ejemplo <code>/dev/mem</code> .
-s	--string <palabra_clave>	Muestra información según la palabra clave: bios-vendor, bios-version, bios-release-date, system-manufacturer, system-product-name, system-version, system-serial-number, system-uuid, baseboard-manufacturer, baseboard-product-name, baseboard-version, baseboard-serial-number, baseboard-asset-tag, chassis-manufacturer, chassis-type, chassis-version, chassis-serial-number, chassis-asset-tag, processor-family, processor-

		manufacturer, processor-version, processor-frequency.
-t	--type <tipo>	Muestra información del tipo mencionado: 0 BIOS 1 System 2 Base Board 3 Chassis 4 Processor 5 Memory Controller 6 Memory Module 7 Cache 8 Port Connector 9 System Slots 10 On Board Devices 11 OEM Strings 12 System Configuration Options 13 BIOS Language 14 Group Associations 15 System Event Log
		16 Physical Memory Array 17 Memory Device 18 32-bit Memory Error 19 Memory Array Mapped Address 20 Memory Device Mapped Address 21 Built-in Pointing Device 22 Portable Battery 23 System Reset 24 Hardware Security 25 System Power Controls

		26 Voltage Probe 27 Cooling Device 28 Temperature Probe 29 Electrical Current Probe 30 Out-of-band Remote Access 31 Boot Integrity Services 32 System Boot 33 64-bit Memory Error 34 Management Device 35 Management Device Component 36 Management Device Threshold Data 37 Memory Channel 38 IPMI Device 39 Power Supply 40 Additional Information 41 Onboard Device Los tipos de 128 a 255 están reservados para datos OEM específicos.

Para reemplazar algunos tipos, contamos con las palabras clave:

Palabra clave	Tipo
bios	0, 13
system	1, 12, 15, 23, 32

baseboard	2, 10, 41
chassis	3
processor	4
memory	5, 6, 16, 17
cache	7
connector	8
slot	9

Las palabras clave no diferencian entre mayúsculas y minúsculas.

Ejemplo de sintaxis

Los cuatro comandos siguientes son similares:

- `dmidecode --type 2 --type 10 --type 41`
- `dmidecode --type 2,10,41`
- `dmidecode --type baseboard`
- `dmidecode --type BASEBOARD`

Mostrar información de la BIOS

La palabra clave bios corresponde a los tipos 0 (bios) y 13 (bios language):

```
# dmidecode -t bios
# dmidecode 2.11
SMBIOS 2.5 present.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
    Vendor: innotek GmbH
    Version: VirtualBox
    Release Date: 12/01/2006

    Address: 0xE0000
```

```
Runtime Size: 128 kB
```

```
ROM Size: 128 kB
```

```
Characteristics:
```

```
    ISA is supported
```

```
    PCI is supported
```

```
    Boot from CD is supported
```

```
    Selectable boot is supported
```

```
    8042 keyboard services are supported (int 9h)
```

```
    CGA/mono video services are supported (int 10h)
```

```
    ACPI is supported
```

La BIOS está fabricada por innotek GmbH porque la máquina está virtualizada en VirtualBox. Habremos obtenido sus características.

5. Bus

Para ver la información acerca de los diferentes buses, contamos con los siguientes comandos:

- `lspci`
- `lsusb`
- `lsscsi`
- `lspcmcia`

Podemos utilizar `lshw` con la opción `-businfo`.

a. `lspci`

El comando `lspci` muestra información acerca de los buses PCI.

`lspci` se instala por defecto en las tres distribuciones. Toma sus datos del archivo `/usr/share/misc/pci.ids` que contiene una lista de identificadores conocidos (fabricantes, periféricos, clases...).

La herramienta update-pciids permite descargar una versión más reciente de este archivo.

update-pciids

Downloaded daily snapshot dated 2016-04-30 03:15:02

Si por el contrario el programa lspci se compila con soporte para compresión, leerá prioritariamente /usr/share/misc/pci.ids.gz antes de leer pci.ids.

Sintaxis

`lspci <opción>`

Opciones útiles:

Opción corta	Descripción
(ninguna)	Muestra una lista corta de dispositivos.
-v	Muestra con detalle (verbose) información de todos los dispositivos.
-vv	Muestra con mayor detalle (verbose) información de todos los dispositivos.
-vvv	Muestra todavía con más detalle (verbose) información de todos los dispositivos.

Mostrar los buses PCI

lspci

00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC

[Natoma] (rev 02)

00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA

[Natoma/Triton II]

00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE

(rev 01)

00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH

VirtualBox Graphics Adapter

00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit

Ethernet Controller (rev 02)

00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox

Guest Service

00:05.0 Multimedia audio controller: Intel Corporation 82801AA
AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW
(ICH6 Family) USB2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM
(ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)

Mostrar una lista más detallada

lspci -v

00:00.0 Host bridge: Intel Corporation Core Processor DMI (rev 11)
Subsystem: Dell Device 02ef
Flags: fast devsel
Capabilities: [40] #00 [0000]

00:03.0 PCI bridge: Intel Corporation Core Processor PCI Express
Root Port 1 (rev 11) (prog-if 00 [Normal decode])
Flags: bus master, fast devsel, latency 0
Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
I/O behind bridge: 0000d000-0000dfff
Memory behind bridge: f6d00000-f6efffff
Prefetchable memory behind bridge: 00000000e0000000-0
0000000effffffff
Capabilities: [40] Subsystem: Dell Device 02ef
Capabilities: [60] MSI: Enable- Count=1/2 Maskable+ 64bit-
Capabilities: [90] Express Root Port (Slot+), MSI 00
Capabilities: [e0] Power Management version 3
Capabilities: [100] Advanced Error Reporting

Capabilities: [150] Access Control Services

Capabilities: [160] Vendor Specific Information: ID=0002

```
Rev=0 Len=00c <?>
```

```
Kernel driver in use: pcieport
```

```
00:08.0 System peripheral: Intel Corporation Core Processor System  
Management Registers (rev 11)
```

```
Flags: fast devsel
```

```
Capabilities: [40] Express Root Complex Integrated Endpoint,  
MSI 00
```

```
Capabilities: [100] Vendor Specific Information: ID=0000
```

```
Rev=0 Len=000 <?>
```

```
...
```

```
3f:05.2 Host bridge: Intel Corporation Core Processor Integrated  
Memory Controller Channel 1 Rank Registers (rev 04)
```

```
Subsystem: Intel Corporation Device 8086
```

```
Flags: bus master, fast devsel, latency 0
```

```
3f:05.3 Host bridge: Intel Corporation Core Processor Integrated  
Memory Controller Channel 1 Thermal Control Registers (rev 04)
```

```
Subsystem: Intel Corporation Device 8086
```

```
Flags: bus master, fast devsel, latency 0
```

Mostrar con detalle información sobre la tarjeta gráfica

Para lograr este objetivo, usamos `lspci -v` con un filtro:

```
# lspci -v | perl -ne '/VGA/..^\$/ and /VGA|Kern/ and print'
```

```
01:00.0 VGA compatible controller: Advanced Micro Devices [AMD]  
nee ATI Broadway XT [Mobility Radeon HD 5870] (prog-if 00 [VGA  
controller])
```

```
Kernel driver in use: radeon
```

b. lsusb

El comando `lsusb` muestra información acerca de los buses USB. La lógica de funcionamiento es similar a la de `lspci`.

lsusb se instala por defecto en las tres distribuciones. Toma sus datos del archivo `/var/lib/usbutils/usb.ids` que contiene una lista de identificadores conocidos.

El archivo `usb.ids` se ubica en la carpeta `/usr/share/hwdata` para la distribución CentOS.

Sintaxis

```
lsusb <opción>
```

Opciones útiles:

Opción corta	Descripción
(ninguna)	Muestra una lista corta de dispositivos.
-t	Crea un árbol de dispositivos USB.
-v	Muestra con detalle (verbose) información de los dispositivos visualizados.

Mostrar el bus

La clase `bus` de `lshw` permite a su vez mostrar información acerca de los buses USB:

```
# lshw -c bus
*-core
  description: Motherboard
  product: VirtualBox
  vendor: Oracle Corporation
  physical id: 0
  version: 1.2
  serial: 0
*-usb
  description: USB controller
  product: KeyLargo/Intrepid USB
  vendor: Apple Inc.

  physical id: 6
  information bus: pci@0000:00:06.0
  version: 00
```

```

width: 32 bits
clock: 33MHz
capabilities: ohci bus_master cap_list
configuration: driver=ohci-pci latency=64
resources: irq:22 memory:f0804000-f0804fff
*-usb:1
description: USB controller
product: 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI
Controller
vendor: Intel Corporation
physical id: b
information bus: pci@0000:00:0b.0
version: 00
width: 32 bits
clock: 33MHz
capabilities: ohci bus_master cap_list
configuration: driver=ohci-pci latency=64
resources: irq:19 memory:f0805000-f0805fff

```

Mostrar una lista corta

```

# lsusb
Bus 001 Device 002: ID 8087:0020 Intel Corp. Integrated Rate
Matching Hub
Bus 002 Device 002: ID 8087:0020 Intel Corp. Integrated Rate
Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 05ca:1815 Ricoh Co., Ltd

Bus 001 Device 004: ID 0a5c:5800 Broadcom Corp. BCM5880 Secure
Applications Processor
Bus 002 Device 003: ID 05ac:1301 Apple, Inc. iPod Shuffle 2.Gen

```

Bus 002 Device 004: ID 1a81:2203 Holtek Semiconductor, Inc. Laser
Gaming mouse

c. lsscsi

El comando lsscsi muestra una lista de dispositivos SCSI. No se instala por defecto en las tres distribuciones mencionadas en este libro.

Debian y Ubuntu

Para realizar su instalación, escriba el siguiente comando:

```
# apt-get install lsscsi
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  lsscsi
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no
actualizados.
Necesito descargar 35,6 kB de archivos.
Se utilizarán 105 kB de espacio de disco adicional después de esta
operación.
Des: 1 http://es.archive.ubuntu.com/ubuntu/ trusty/main lsscsi
amd64 0.27-2 [35,6 kB]
Descargados 35,6 kB en 0seg. (115 kB/s)
Seleccionando el paquete lsscsi previamente no seleccionado.
(Leyendo la base de datos ... 199483 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../lsscsi_0.27-3_amd64.deb ...
Unpacking lsscsi (0.27-2) ...
Procesando disparadores para man-db (2.7.0.2-5)

...Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Configurando lsscsi (0.27-2) ...
```

CentOS

La instalación se realiza con yum install:

```
# yum -y install lsscsi
Complementos cargados: fastestmirror, langpacks
base | 3.6 kB 00:00
extras | 3.4 kB 00:00
updates | 3.4 kB 00:00
Loading mirror speeds from cached hostfile
* base: mir01.syntis.net
* extras: mir01.syntis.net
* updates: mir01.syntis.net
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete lsscsi.x86_64 0:0.27-3.el7 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package           Arquitectura  Versión           Repositorio      Tamaño
=====
Instalando:
  lsscsi           x86_64       0.27-3.el7       base             47 k

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 47 k 47 k

Tamaño instalado: 88 k
Downloading packages:
lsscsi-0.27-3.el7.x86_64.rpm | 47 kB 00:00:00
Running transaction check
```

Running transaction test

Transaction test succeeded

Running transaction

Instalando : lsscsi-0.27-3.el7.x86_64 1/1

Comprobando : lsscsi-0.27-3.el7.x86_64 1/1

Instalado:

lsscsi.x86_64 0:0.27-3.el7

¡Listo!

Sintaxis

lsscsi <opciones>

Opciones necesarias:

Opción		Descripción
corta	larga	
(ninguna)	(ninguna)	Ver resumen de los dispositivos SCSI.
-d	--device	Después de la visualización del nombre del dispositivo SCSI, se muestran entre corchetes el número principal y el número menor del dispositivo. Por ejemplo: /dev/sda [8:0].
-g	--generic	Muestra el nombre del dispositivo genérico SCSI (sg).

Ver resumen de los dispositivos SCSI

lsscsi

[0:0:0:0]	disk	ATA	TOSHIBA	MK5056GS	LJ00	/dev/sda
[1:0:0:0]	cd/dvd	HL-DT-ST	DVD+-RW	GS20N	A110	/dev/sr0
[2:0:0:0]	disk	ATA	TOSHIBA	MK5056GS	LJ00	/dev/sdb

Vista más detallada

```
# lsscsi -dg
[0:0:0:0]    disk      ATA      TOSHIBA MK5056GS LJ00  /dev/sda
[8:0]       /dev/sg0 [21:0]
[1:0:0:0]    cd/dvd   HL-DT-ST DVD+-RW GS20N   A110  /dev/sr0
[11:0]      /dev/sg1 [21:1]
[2:0:0:0]    disk      ATA      TOSHIBA MK5056GS LJ00  /dev/sdb
[8:16]      /dev/sg2 [21:2]
[16:0:0:0]   disk      Apple    iPod          2.70  /dev/sdc
[8:32]      /dev/sg3 [21:3]
```

d. lspcmcia

Debian, Ubuntu Server y CentOS no incluyen por defecto este comando. Es cierto que PCMCIA es muy útil en una distribución orientada a una estación de trabajo que está instalada en un ordenador portátil.

Debian y Ubuntu Server

La instalación se realiza añadiendo el paquete pcmciautils:

```
# apt-get -y install pcmciautils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 pcmciautils
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 32,6 kB de archivos.
Se utilizarán 124 kB de espacio de disco adicional después de esta
operación.
Des: http://ftp.es.debian.org/debian/ jessie/main pcmciautils amd64
```

018-8 [32,6 kB]

Descargados 32,6 kB en 0s (645 kB/s)

Seleccionando el paquete pcmciautils previamente no seleccionado.

(Leyendo la base de datos ... 247275 ficheros o directorios instalados actualmente.)

Preparando para desempaquetar .../pcmciautils_018-8_amd64.deb ...

Desempaquetando pcmciautils (018-8) ...

Procesando disparadores para man-db (2.7.0.2-5) ...

Configurando pcmciautils (018-8) ...

CentOS

El paquete pcmciautils ya no se encuentra en esta distribución.

Sintaxis

```
lspcmcia <opciones>
```

Opciones útiles:

Opción corta	Descripción	Mostrar los dispositivos PCMCIA
(ninguna)	Muestra una lista corta de los dispositivos PCMCIA.	# lspcmcia Socket 0 Bridge : [yenta_cardbus]] (bus ID: 0000: 03:01.0)
-v	Muestra en detalle (verbose) la información.	

Ver la información con más detalle

```
# lspcmcia -v
```

```
Socket 0 Bridge: [yenta_cardbus] (bus ID: 0000:03:01.0)
```

```
Configuration: state: on ready: yes
```

6. Memoria

a. Información acerca de la memoria

El archivo `/proc/meminfo` devuelve bastante información sobre la memoria de un equipo. También contamos con la clase `memory` de `lshw`, que muestra las características de la memoria caché y los bancos de memoria:

```
# lshw -c memory
*-memory
  description: System memory
  physical id: 0
  size: 1504MiB
```

b. Diagnosticar la memoria

Una máquina puede volverse inestable porque los conectores de memoria son de mala calidad o están defectuosos. Memtest86+ es una utilidad que realiza un diagnóstico completo de la memoria de un equipo.

Desde el sitio web oficial <http://www.memtest.org/>, descargamos la imagen ISO de Memtest86+. Tenemos dos posibilidades:

- Crear una llave USB arrancable (bootable). Deberemos elegir en el sitio **Download - Auto-installer for USB Key (Win 9X/2k/XP/7)**.
- Crear un live CD. En este caso, seleccionamos **Download - Pre-Compiled Bootable Binary (.gz)**. El archivo `memtest86+-4.20.bin.gz` está comprimido con `gzip`. En caso contrario, se ofrece otra descarga: **Download - Pre-Compiled Bootable Binary (.zip)**. Está comprimido en formato `zip`. Una vez efectuada la descompresión, grabe la imagen en un CD.
- Arranca el ordenador desde el CD o la llave USB. Memtest86+ se ejecuta para realizar su diagnóstico:

```

Memtest86+ v4.20 | Pass100% #####
Intel Core Gen2 3393 MHz | Test 19% #####
L1 Cache: 32K | Test #8 [Modulo 20, Random pattern]
L2 Cache: 256K 51403 MB/s | Testing: 196K - 1536M 1536M
L3 Cache: 8192K 38552 MB/s | Pattern: 4b1a5c0d-3
Memory : 1536M 13847 MB/s |-----
IMC : Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz / BCLK : 242 MHz
Settings: RAM : 1001MHz (DDR3-2003) / CAS : 7-12-1-68 / Single Channel

WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors  ECC Errs
-----
0:16:15  1536M      0K      e820    on   off   Std    1      0

*****Pass complete, no errors, press Esc to exit*****
(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock

```

Si obtenemos líneas rojas en la pantalla, tenemos por seguro una tarjeta de memoria defectuosa.

```

WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors  ECC E
-----
1:49:11  128G      5928K      e820    on   off   Std    0  1281280

Tst  Pass  Failing Address  Good  Bad  Err-Bits  Count Cha
---
7    0  000085e36c0 - 133.8MB 13f5e32a ffffffff ec0d1cd5 1281460
7    0  000085e36d0 - 133.8MB 5e353253 ffffffff 5d8c73e3 1281463
7    0  000085e36e0 - 133.8MB 7016a154 ff7fffffff 8fb45ea6 1281465
7    0  000085e36e8 - 133.8MB ec7e0761 fdfffffff 3325782e 1281463
7    0  000085e36f4 - 133.8MB 36c64e7e efffffff d939b181 1281470
7    0  000085e36f0 - 133.8MB 532b7c35 bfffffff da3483fa 1281473
7    0  000085e3708 - 133.8MB f3347933 ffffffff 0ccb86ce 1281475
7    0  000085e3714 - 133.8MB c58c98a1 ffffffff 38236555 1281473
7    0  000085e371c - 133.8MB 835ab3c4 ffffffff 7ca54c7b 1281480

```

Debemos apagar el equipo y sustituir la tarjeta de memoria afectada.

7. Tarjeta gráfica

Para identificar la tarjeta gráfica, utilizamos lshw con la clase video:

```
#lshw -c video
*-display
  description: VGA compatible controller
  product: SVGA II Adapter
    vendor: VMWare
    physical id: f
    bus info: pci@0000:00:0f.0
    version: 00
    width: 64 bits
    clock: 33MHz
capabilities: vga_controller bus_master
cap_list rom
  configuration: driver=vmwgfx latency=64
  resources: irq:16 ioport:1070(size=16)
memory:e8000000-ffffffff
memory:fe000000-fe7fffff memory:60000000-60007fff
```

El equipo del ejemplo utiliza la tarjeta gráfica virtual empleada por VMware. El driver utilizado es vmwgfx.

8. Discos duros

a. Información sobre los controladores

La clase storage del comando lshw muestra información sobre los controladores de disco:

```
# lshw -c storage
*-ide
    description: IDE interface
    product: 82371AB/EB/MB PIIX4 IDE
    vendor: Intel Corporation
    physical id: 7.1
    bus info: pci@0000:00:07.1
    version: 01
    width: 32 bits
    clock: 33MHz
    capabilities: ide bus_master
    configuration: driver=ata_piix latency=64
    resources: irq:0 ioport:1f0(size=8) ioport:3f6 ioport:170(size=8)
ioport:376 ioport:1060(size=16)
*-scsi
    description: SCSI storage controller
    product: 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI
    vendor: LSI Logic / Symbios Logic
    physical id: 10
    bus info: pci@0000:00:10.0
    logical name: scsi2
    version: 01
    width: 64 bits
    clock: 33MHz
    capabilities: scsi bus_master cap_list rom
    configuration: driver=mptspi latency=64 maxlatency=255 mingnt=6
    resources: irq:17 ioport:1400(size=256) memory:feba0000-febbffff
memory:febc0000-febdffff memory:60008000-6000bfff
```

```
*-scsi
    physical id: 2
    logical name: scsi1
    capabilities: emulated
```

b. Información de los discos

La clase disk del comando lshw muestra información de los discos duros y lectores de CD o DVD:

```
# lshw -c disk
*-disk
    description: ATA Disk
    product: TOSHIBA MK5056GS
    vendor: Toshiba
    physical id: 0.0.0
    bus info: scsi@0:0.0.0
    logical name: /dev/sda
    version: LJ00
    serial number: 11PHT6NFT
    size: 465GiB (500GB)
    capabilities: partitioned partitioned:dos
    configuration: ansiversion=5 sectorsize=512
signature=0faa21d8
*-cdrom
    description: DVD-RAM writer
    product: DVD+-RW GS20N
    vendor: HL-DT-ST
    physical id: 0.0.0
    bus info: scsi@1:0.0.0
    logical name: /dev/cdrom
    logical name: /dev/cdrw
    logical name: /dev/dvd
    logical name: /dev/dvdrw
```

```
logical name: /dev/sr0
version: A110
capabilities: removable audio cd-r cd-rw dvd dvd-r dvd-ram
configuration: ansiversion=5 status=open

*-disk
description: ATA Disk
product: TOSHIBA MK5056GS
vendor: Toshiba
physical id: 0.0.0
information bus: scsi@2:0.0.0
logical name: /dev/sdb
version: LJ00
numéro de série: 11PHT6NLT
size: 465GiB (500GB)
capabilities: partitioned partitioned:dos
configuration: ansiversion=5 sectorsize=512
signature=0006d92b
```

c. Información de los volúmenes

La clase volume del comando lshw muestra información de los volúmenes y de los sistemas de archivo utilizados:

```
# lshw -c volume

*-volume:0
description: *-volume:0
description: Linux filesystem partition
vendor: Linux
physical id: 1
bus info: scsi@0:0.0.0,1
logical name: /dev/sda1
logical name: /boot
version: 1.0

serial: 3daed7e0-2542-4ab5-b760-313ccdab3b1f
```



```
size: 953MiB
capacity: 953MiB
capabilities: primary bootable extended_attributes large_files
ext2 initialized
configuration: filesystem=ext2 lastmountpoint=/boot
modified=2016-05-08 10:09:55 mount.fstype=ext2 mount.options=rw,
relatime mounted=2016-05-08 10:09:55 state=mounted
*-volume:1
  description: Extended partition
  physical id: 2
  bus info: scsi@0:0.0.0,2
  logical name: /dev/sda2
  size: 7236MiB
  capacity: 7236MiB
  capabilities: primary extended partitioned partitioned:extended
*-logicalvolume
  description: Linux LVM Physical Volume partition
  physical id: 5
  logical name: /dev/sda5
  serial: ZNaQIt-NUfZ-UOPE-Ymeb-N2sX-wbdo-GWlATW
  size: 7236MiB
  capacity: 7236MiB
  capabilities: multi lvm2
*-volume:2
  description: Volume EXT4
  physical id: 3
  bus info: scsi@0:0.0.0,3
  logical name: /dev/sda3
  version: 1.0
  serial: 8622251d-6c1f-4962-8840-9335ec8b7595
  size: 9537MiB
  capacity: 9537MiB
```

```

capabilities: primary journaled extended_attributes
*-volume
description: Linux LVM Physical Volume partition
physical id: 1
bus info: scsi@1:0.0.0,1
logical name: /dev/sdb1
serial: cdPzp6-3Ptn-Xtxy-7u1N-xx7c-DqFh-Fmeo0k
size: 10238MiB
capacity: 10238MiB

capabilities: primary journaled extended_attributes
large_files huge_files dir_nlink recover extents ext4 ext2
initialized
configuration: created=2013-05-30 20:06:04 filesystem=ext4
lastmountpoint=/home modified=2013-08-10 22:26:07
mount.fstype=ext4 mount.options=rw,relatime,data=ordered
mounted=2013-08-10 22:26:07 state=mounted

```

Instalación en CentOS

El paquete smartmontools ya viene instalado.

Activación de smart

La herramienta smartctl controla y supervisa los discos SMART.

Sintaxis

```
smartctl <opciones> <disco>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-s <valor>	--smart=<valor>	Activa (valor on) o desactiva (valor off) SMART en el dispositivo.

-o	--offlineauto=<valor>	Activa (valor on) o bien desactiva (valor off) la prueba «offline» automática que se realiza cada 4 horas.
-S	--saveauto=<valor>	Activa (valor on) o bien desactiva (valor off) la copia de seguridad automática SMART de los atributos específicos del proveedor del dispositivo.

El valor <disco> se reporta como sigue: /dev/sda.

Activación con éxito

```
# smartctl --smart=on --offlineauto=on --saveauto=on /dev/sdb
smartctl 5.43 2012-06-30 r3573 [x86_64-linux-3.8.0-27-generic]
(local build)
Copyright (C) 2002-12 by Bruce Allen,
http://smartmontools.sourceforge.net

=== START OF ENABLE/DISABLE COMMANDS SECTION ===
SMART Enabled.
SMART Attribute Autosave Enabled.
SMART Automatic Offline Testing Enabled every four hours.
```

Activación fallida

```
# smartctl --smart=on --offlineauto=on --saveauto=on /dev/sda
smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]
(local build)
Copyright (C) 2002-14, Bruce Allen, Christian Franke,
www.smartmontools.org
http://smartmontools.sourceforge.net

SMART support is: Unavailable - device lacks SMART capability.
A mandatory SMART command failed: exiting. To continue, add one or
more '-T permissive' options
```

El dispositivo no permite la activación de SMART. Es un disco virtual generado por una máquina virtual de Oracle VM VirtualBox.

Recuperación de información

```
# smartctl -a /dev/sdb

smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]
(local build)
Copyright (C) 2002-14, Bruce Allen, Christian Franke,
www.smartmontools.org
http://smartmontools.sourceforge.net

=== START OF INFORMATION SECTION ===
Model Family:      Toshiba 2.5" HDD MK..56GSY
Device Model:      TOSHIBA MK5056GSYF
Serial Number:     11PHT6NLT
LU WWN Device Id:  5 000039 301805a6f
Firmware Version:  LJ001D
User Capacity:     500 107 862 016 bytes [500 GB]
Sector Size:       512 bytes logical/physical
Device is:         In smartctl database [for details use: -P show]
ATA Version is:    8
ATA Standard is:   Exact ATA specification draft version not indicated
Local Time is:     Tue Aug 13 12:24:19 2013 CEST
SMART support is:  Available - device has SMART capability.
SMART support is:  Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x80)  Offline data collection
                                   activity was never started.
                                   Auto Offline Data Collection: Enabled.
Self-test execution status:      (   0)   The previous self-test
                                   routine completed
```

```

                                without error or no self-test has
                                ever been run.

Total time to complete Offline
data collection:                ( 120) seconds.
Offline data collection
capabilities:                    (0x5b) SMART execute Offline immediate.
                                Auto Offline data collection on/off
                                support.
                                Suspend Offline collection upon
                                new command.
                                Offline surface scan supported.
                                Self-test supported.
                                No Conveyance Self-test supported.
                                Selective Self-test supported.
SMART capabilities:             (0x0003)  Saves SMART data before
                                entering power-saving mode.
                                Supports SMART auto save timer.
Error logging capability:       (0x01)  Error logging supported.
                                General Purpose Logging supported.

Short self-test routine
recommended polling time:       (  2) minutes.
Extended self-test routine
recommended polling time:       ( 126) minutes.
SMART capabilities:             (0x0039)
                                SCT Status supported.
                                SCT Error Recovery Control supported.
                                SCT Feature Control supported.
                                SCT Data Table supported.


SMART Attributes Data Structure revision number: 128
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE

```

UPDATED WHEN_FAILED RAW_VALUE

1	Raw_Read_Error_Rate	0x000b	100	100	050	Pre-fail
Always	-	0				
3	Spin_Up_Time	0x0027	100	100	001	Pre-fail
Always	-	2127				
5	Reallocated_Sector_Ct	0x0033	100	100	050	Pre-fail
Always	-	0				
9	Power_On_Minutes	0x0032	077	077	000	Old_age
Always	-	9249h+01m				
12	Power_Cycle_Count	0x0032	100	100	000	Old_age
Always	-	2249				
191	G-Sense_Error_Rate	0x0032	100	100	000	Old_age
Always	-	65				
192	Power-Off_Retract_Count	0x0032	100	100	000	Old_age
Always	-	37				
193	Load_Cycle_Count	0x0032	096	096	000	Old_age
Always	-	47897				
194	Temperature_Celsius	0x0022	100	100	000	Old_age
Always	-	41 (Min/Max 6/58)				
199	UDMA_CRC_Error_Count	0x0032	100	100	000	Old_age
Always	-	9509218				
200	Multi_Zone_Error_Rate	0x0032	100	100	000	Old_age
Always	-	17841953				
240	Head_Flying_Hours	0x0032	096	096	000	Old_age
Always	-	99837				
241	Total_LBAs_Written	0x0032	100	100	000	Old_age
Always	-	70697682754				
242	Total_LBAs_Read	0x0032	100	100	000	Old_age
Always	-	235632662681				
254	Free_Fall_Sensor	0x0032	100	100	000	Old_age
Always	-	2				

SMART Error Log Version: 1

No Errors Logged

SMART Self-test log structure revision number 1

Num	Test_Description	Status	Remaining	LifeTime(hours)
LBA_of_first_error				
# 1	Short offline	Completed without error	00%	7176
	-			
# 2	Short offline	Completed without error	00%	4769
	-			
# 3	Short offline	Completed without error	00%	0
	-			

SMART Selective self-test log data structure revision number 1

SPAN	MIN_LBA	MAX_LBA	CURRENT_TEST_STATUS
1	0	0	Not_testing
2	0	0	Not_testing
3	0	0	Not_testing
4	0	0	Not_testing
5	0	0	Not_testing

Selective self-test flags (0x0):

After scanning selected spans, do NOT read-scan remainder of disk.

If Selective self-test is pending on power-up, resume after 0
minute delay.

Test

Sintaxis

```
smartctl <opciones> <disco>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-t <valor>	--test=<valor>	El valor short efectúa una prueba corta en segundo plano. El valor long efectúa una prueba larga en segundo plano.
-X	--abort	Abandona la prueba en curso.
-l <tipo>	--log=<tipo>	Si el tipo es error, muestra los errores provenientes de la prueba. Si el tipo es selftest, muestra el registro de la funcionalidad selftest.
-q <valor>	--quietmode=<valor>	Si el valor es errorsonly entonces solo se muestran los errores.
-H	--health	El dispositivo devuelve su estado SMART.

El valor <disco> se reporta como sigue: /dev/sda.

Realizar un test corto

```
# smartctl -t short /dev/sdb
smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]
(local build)
Copyright (C) 2002-14, Bruce Allen, Christian Franke,
www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Short self-test routine
immediately in off-line mode".
Drive command "Execute SMART Short self-test routine immediately
in off-line mode" successful.
```



```
Testing has begun.  
Please wait 2 minutes for test to complete.  
Test will complete after Tue Aug 13 12:39:25 2013  
  
Use smartctl -X to abort test.
```

El test concluirá en dos minutos.

Realizar un test largo

```
# smartctl -t long /dev/sdb  
smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]  
(local build)  
Copyright (C) 2002-14, Bruce Allen, Christian Franke,  
www.smartmontools.org  
  
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===  
Sending command: "Execute SMART Extended self-test routine  
immediately in off-line mode".  
Drive command "Execute SMART Extended self-test routine  
immediately in off-line mode" successful.  
Testing has begun.  
Please wait 126 minutes for test to complete.  
Test will complete after Tue Aug 13 15:14:51 2013  
  
Use smartctl -X to abort test.
```

El test concluirá en 126 minutos.

Abortar un test

```
# smartctl -X /dev/sdb  
smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]  
(local build)  
Copyright (C) 2002-14, Bruce Allen, Christian Franke,  
www.smartmontools.org
```

```
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
```

Sending command: "Abort SMART off-line mode self-test routine".

Self-testing aborted!

Acceso a los resultados de los tests

Pasado el tiempo indicado en el arranque de smartctl, escribimos:

```
# smartctl -l selftest /dev/sdb
smartctl 6.4 2014-10-07 r4002 [x86_64-linux-3.16.0-4-amd64]
(local build)
Copyright (C) 2002-14, Bruce Allen, Christian Franke,
www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Self-test log structure revision number 1
Num  Test_Description      Status                    Remaining
LifeTime(hours)  LBA_of_first_error
# 1  Short offline          Completed without error     00%
9249              -
# 2  Short offline          Completed without error     00%
7176              -
# 3  Short offline          Completed without error     00%
4769              -
# 4  Short offline          Completed without error     00%
0                -
```

Acceder solo a los errores

```
# smartctl -q erroronly -H -l selftest /dev/sdb
```

En este caso, el disco no tiene errores, ya que smartctl no devuelve nada.

9. Tarjeta de red

La clase network del comando lshw muestra información sobre las interfaces de red:

```
# lshw -c network

*-network
    descripción: Ethernet interface
    producto: RTL8111/8168/8411 PCI Express Gigabit Ethernet
Controller
    fabricante: Realtek Semiconductor Co., Ltd.
    id físico: 0
    información del bus: pci@0000:07:00.0
    nombre lógico: eno1
    versión: 06
    serie: 2c:41:38:5d:5c:90
    tamaño: 10Mbit/s
    capacidad: 1Gbit/s
    anchura: 64 bits
    reloj: 33MHz
    capacidades: pm msi pciexpress msix vpd bus_master cap_list
ethernet physical tp mii 10bt 10bt-fd 100bt 100bt-fd 1000bt
1000bt-fd autonegotiation
    configuración: autonegotiation=on broadcast=yes driver=r8169
driverversion=2.3LK-NAPI duplex=half firmware= rtl8168e-3_0.0.4
03/27/12 latency=0 link=no multicast=yes port=MII speed=10Mbit/s
    recursos: irq:34 ioport:4000(size=256)
memoria:c0404000-c0404fff memoria:c0400000-c0403fff

*-network
    descripción: Network controller
    producto: BCM4313 802.11bgn Wireless Network Adapter
    fabricante: Broadcom Corporation
    id físico: 0
    información del bus: pci@0000:0d:00.0
```

```
versión: 01
anchura: 64 bits
reloj: 33MHz
capacidades: pm msi pciexpress bus_master cap_list
configuración: driver=bcma-pci-bridge latency=0
recursos: irq:17 memoria:c4500000-c4503fff
*-network
  descripción: Interfaz inalámbrica
  id físico: 1
  nombre lógico: wlp13s0b1
  serie: ac:81:12:b9:a5:c0
  capacidades: ethernet physical wireless
  configuración: broadcast=yes driver=brcmsmac
driverversion=4.4.0-21-generic firmware=610.812 ip=192.168.1.24
link=yes multicast=yes wireless=IEEE 802.11bgn
```

El equipo posee dos interfaces:

- Una es inalámbrica (wireless): una tarjeta BCM4313 de Broadcom con una dirección MAC ac:81:12:b9:a5:c0. Su nombre lógico es /dev/wlp13s0b1.
- La otra es cableada: una RTL8111/8168/8411 PCI Express Gigabit de Realtek con una dirección MAC 2c:41:38:5d:5c:90. Su nombre lógico es /dev/eno1.

Tarjeta de red sin controlador de dispositivo

El equipo cuenta con una tarjeta de red Intel. El comando ifconfig muestra solo la interfaz de loopback, siendo lo su nombre lógico:

```
# ifconfig
lo  Link encap:Bucle local
    inet adr:127.0.0.1  Mask:255.0.0.0
    adr inet6: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:12 errors:0 dropped:0 overruns:0 frame:0
TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmisión:0
RX bytes:836 (1.6 KiB) TX bytes:836 (1.6 KiB)
```

La visualización de la clase network con lshw nos devuelve esto:

```
# lshw -c network
*-network UNCLAIMED
    description: Ethernet controller
    product: 82540EM Gigabit Ethernet Controller
    vendor: Intel Corporation
    physical id: 3
    bus info: pci@0000:00:03.0
    version: 02
    width: 32 bits
    clock: 66MHz
    capabilities: pm pcix cap_list
    configuration: latency=64 mingnt=255
    resources: memory:f0000000-f001ffff ioport:d010(size=8)
```

La palabra UNCLAIMED a la derecha de la palabra network en la segunda línea, significa que la interfaz de red Intel 82540EM Gigabit Ethernet Controller se ha detectado. Sin embargo, no se ha asignado ningún controlador de dispositivo.

No cuenta con un nombre lógico como eth0 ya que este no se indica. La comprobación puede realizarse a su vez de esta manera:

```
# ifconfig eth0
eth0: error fetching interface information: Device not found
```



Recuerde que las interfaces de comunicación de la distribución CentOS 7 tienen como nombre lógico enpXsY, donde X es el número del puerto e y el número de la ubicación (slot).

Como conclusión, el hardware se ha detectado de forma correcta por el sistema. No falta más que la instalación del controlador compatible con el dispositivo, que sin duda es un módulo del kernel a cargar. Se llama e1000 para las tarjetas Intel Pro Gigabit.

```
# modprobe e1000
```

Verificación de la carga:

```
# lsmod |grep e1000
e1000      90558      0
```

El módulo se encuentra cargado de forma correcta. La tarjeta de red tiene como nombre lógico eth0.

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:1a:48
          inet adr:192.168.1.204  Bcast:192.168.1.255
Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fe1a:1a48/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:147 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:13308 (12.9 KiB)  TX bytes:19770 (19.3 KiB)
```

Tarjeta con un controlador de dispositivo

Tenemos el mismo equipo que en el ejemplo anterior, con una tarjeta de red Intel.

El resultado del comando ifconfig indica que las interfaces eth0 y lo están presentes:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:db:fb:ad
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fedb:fbad/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:1240 (1.2 KiB)  TX bytes:6521 (6.3 KiB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 lg file transmisión:0  
RX bytes:1664 (1.6 KiB) TX bytes:1664 (1.6 KiB)
```

Mostrar, con lshw, el hardware de la clase network:

```
# lshw -c network  
*-network  
    description: Ethernet interface  
    product: 82540EM Gigabit Ethernet Controller  
    vendor: Intel Corporation  
    physical id: 3  
    bus info: pci@0000:00:03.0  
    logical name: eth0  
    version: 02  
    serial: 08:00:27:dbb:f:ad  
    size: 1GB/s  
    capacity: 1GB/s  
    width: 32 bits  
    clock: 66MHz  
    capabilities: pm pcix bus_master cap_list ethernet physical  
tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation  
    configuration: autonegotiation=on broadcast=yes driver=e1000  
driverversion=7.3.21-k5-NAPI duplex=full firmware=N/A ip=10.0.2.15  
latency=64 link=yes mingnt=255 multicast=yes port=twisted pair  
speed=1GB/s  
    resources: irq:19 memory:f0000000-f001ffff  
ioport:d010(size=8)
```

Por un lado, la interfaz de red Intel 82540EM Gigabit Ethernet Controller no se encuentra UNCLAIMED, y a su vez cuenta con el nombre lógico eth0.

El controlador de dispositivo e1000 se encuentra bien e instalado de forma correcta ya que podemos leer Driver=e1000.

Ejercicio

1. Inicie una sesión en una consola como el usuario root.
2. Compruebe la presencia del comando lshw. Si no se encuentra, instálelo.
3. Identifique el nombre de la interfaz de comunicación cableada con lshw y el nombre del controlador del dispositivo.
4. Visualice las entradas del registro del controlador.
5. Verifique que el controlador está cargado.
6. Descargue el módulo del kernel que controla esta interfaz.
7. Compruebe con lshw si la interfaz de comunicación no ha dejado un nombre lógico o un controlador.
8. Descargue el módulo del kernel que controla esta interfaz.

Introducción

Cuando instalamos una distribución de Linux y ejecutamos un particionado manual, podemos elegir entre el LVM (Logical Volume Manager) y las particiones tradicionales.

Si seleccionamos un particionado asistido, la mayoría de las distribuciones usan LVM por defecto. Sin embargo, esta opción se adapta bien a una máquina de formación, por ejemplo. En producción, es preferible particionar manualmente.

Particiones

Una partición es una parte de un disco duro destinada a albergar un sistema de archivos como ext4. La principal razón para crear varias particiones es la seguridad. Si una partición se corrompe, las demás permanecerán en buen estado.

Nada nos impide poner el sistema, aplicaciones y datos en una única partición montada como /. Pero recuerde que siempre debe tener una partición swap. En efecto, para instalar Linux, son necesarias dos particiones como mínimo:

- una para la raíz /. El sistema básico no excede los 8 GB.
- una para el swap. Su tamaño depende del uso del ordenador, de la cantidad de RAM y del espacio en disco disponible con el que contemos.

He aquí algunas recomendaciones sobre el tamaño del swap:

RAM	Tamaño del swap
< 1 GB	Doble de la memoria RAM
Entre 1 y 4 GB	Un mínimo de 2 GB
Entre 4 y 16 GB	Un mínimo de 4 GB
Entre 16 y 64 GB	Un mínimo de 8 GB
Entre 64 y 256 GB	Un mínimo de 16 GB
Entre 256 y 512 GB	Un mínimo de 32 GB

Si empleamos una estación de trabajo Linux en un ordenador portátil, el tamaño del swap debería ser al menos equivalente al de la memoria RAM para satisfacer las necesidades de la hibernación.

El modo de particionar varía de una máquina a otra en función de su uso.

Particionado de una estación de trabajo

Una estación de trabajo necesita por lo general cuatro particiones:

Partición	Tamaño recomendado	Descripción
swap	Entre 2 y 8 GB	Depende del tamaño de la RAM. Consulte la primera tabla del capítulo.
/	10 GB	El sistema no excede por lo general los 8 GB de ocupación de espacio en disco.
/home	El mayor tamaño posible	En función de los datos de usuario.
/opt	5 GB	Aplicaciones que no proceden de la distribución.
/var	3 GB como mínimo	Contiene los archivos de datos que se pueden modificar con frecuencia, por lo tanto variables.

Particionado de un servidor web

Un servidor Apache HTTPD Server necesita particiones:

Partición	Tamaño recomendado	Descripción
swap	Entre 2 y 32 GB	Depende del tamaño de la RAM. Consulte la primera tabla del capítulo.
/boot	500 MB	Software de arranque GRUB y el kernel de Linux.
/	10 GB	Sistema
/tmp	2 GB	Archivos temporales.
/var	5 GB	Datos del sistema, en particular, los registros.
/opt/httpd	5 GB	Ubicación de Apache Httpd Server. Una versión compilada por nuestra parte en lugar de utilizar la que se proporciona

		con la distribución (lo volveremos a comentar en el capítulo Mantenimiento de las aplicaciones).
/websites	El mayor tamaño posible	Alojamiento de los servidores virtuales.

1. Tabla de particiones

Cuando creamos una partición en Debian y Ubuntu debemos indicar el tipo de partición: aix, amiga, bsd, dvh, gpt, mac, msdos, pc98, sun, loop.

a. Tipo msdos

La tabla de partición de tipo msdos es el particionado tradicional de los PC. Contiene un MBR (Master Boot Record), o en español un sector de arranque, que es el primer sector del disco duro, es decir, el cilindro 0, cabeza 0 y el sector 1. Su tamaño es de 512 bytes y contiene la tabla de partición principal y el gestor de arranque (bootloader).

Tamaño en bytes	Descripción
440	Programa de arranque (boot) ejecutado por la BIOS.
6	Firma MBR.
64	Tabla de cuatro particiones (16 bits x 4): • 4 primarias; • 3 particiones primarias y 1 partición extendida.
2	Firma Boot MBR 0xAA55.

El tamaño de las particiones está limitado a 2,2 TB (241 bytes).

b. Tipo GPT

Linux funciona también con una tabla de partición de tipo GPT (GUID partition table) en las plataformas: x86-64, IA-64 y x86. El kernel debe compilarse con la opción CONFIG_EFI_PARTITION, como ocurre en la mayoría de distribuciones.

Con UEFI (Unified Extensible Firmware Interface), el modelo de tabla de partición es de tipo GPT y ofrece dos ventajas concretas:

- El número máximo de particiones aumenta por defecto a 128. Sin embargo, bastaría con aumentar el tamaño de la tabla de particiones para obtener más particiones.
- El tamaño de las particiones puede llegar hasta 9,4 ZB (273 bytes).

Una tabla de particiones GUID utiliza GUID (Globally Unique Identifier) o en el caso de Linux UUID (Universal Unique Identifier) definidos en la RFC 4122 para declarar de manera única las particiones y sus tipos.

El comando blkid muestra los atributos de los dispositivos de bloque:

```
# blkid
/dev/mapper/SystemVG-swapLV: UUID="e22dd157-506c-4726-962b-64d4f1b425a9" TYPE="swap"
/dev/sda1: UUID="OyaFrT-Hoa8-1L8z-fnTY-eyJt-Utf7-Hs3aw2"
TYPE="LVM2_member"
/dev/mapper/SystemVG-rootLV: UUID="ac18424b-581b-4219-a870-65a12955c8a0" TYPE="ext4"
/dev/mapper/SystemVG-homeLV: UUID="9d6ca473-1797-4079-a5e5-8322a3c602fc" TYPE="ext4"
/dev/mapper/SystemVG-optLV: UUID="ae8613bb-e836-432d-8f19-b1a818e50d49" TYPE="ext4"
/dev/sdb: UUID="RVk0aN-x0q0-tJiz-Ld0y-SYm0-hriu-3QLEEC"
TYPE="LVM2_member"
/dev/sdc: UUID="gXJv6W-2UrM-v190-9hKb-hDFF-qLyQ-G3TRbU"
TYPE="LVM2_member"
/dev/sdd: UUID="5iT1z0-Uflq-WieC-nHGx-tfX4-66Ji-AoQq4t"
TYPE="LVM2_member"
/dev/sr0: LABEL="VBOXADDITIONS_4.2.16_86992" TYPE="iso9660"
/dev/mapper/officeVG-docLV: UUID="5c3fd277-1c89-44e6-b342-8137731750b2" TYPE="ext4"
```

c. Convertir de MBR a GPT

La conversión puede realizarse con el comando `gdisk`, una herramienta similar a `fdisk` que soporta ambos tipos, `msdos` y `GPT`. Durante esta operación, todas las particiones y discos lógicos se convierten en particiones `gpt` con sus respectivos `UUID`.

Cabe señalar que existe una tabla secundaria `GPT` al final del disco. Debemos asegurarnos de que el último medio byte del disco no está utilizado.

Por defecto, `gdisk` no está instalado en los sistemas `Debian` y `Ubuntu`. Ya se encuentra en `CentOS`.

Instalación en `Debian` y `Ubuntu`

La instalación del paquete `gdisk` se realiza así:

```
# apt-get -y install gdisk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma
automática y ya no son necesarios.
  diffstat gettext intltool-debian libapt-pkg-perl libarchive-zip-perl
  libasprintf-dev libauthen-sasl-perl libautodie-perl libdigest-
hmac-perl
  libdpkg-perl libemail-valid-perl libfile-fcntllock-perl
libgettextpo-dev
  libgettextpo0 libio-pty-perl libio-socket-inet6-perl libio-
socket-ssl-perl
  libipc-run-perl libipc-system-simple-perl liblist-moreutils-perl
  libmailtools-perl libnet-dns-perl libnet-domain-tld-perl
libnet-ip-perl
  libnet-libidn-perl libnet-smtp-ssl-perl libnet-ssleay-perl
  libperlio-gzip-perl libsocket6-perl libsub-identify-perl
  libtext-levenshtein-perl liburi-perl patchutils t1utils
Use 'apt-get autoremove' to remove them.
Se instalarán los siguientes paquetes NUEVOS:
  gdisk
0 actualizados, 1 se instalarán, 0 para eliminar y 4 no actualizados.
```

```
Necesito descargar 185 kB de archivos.
Se utilizarán 761 kB de espacio de disco adicional después de esta
operación.
Des:1 http://es.archive.ubuntu.com/ubuntu/ trusty-updates/main gdisk
amd64 0.8.8-1ubuntu0.1 [185 kB]
Descargados 185 kB en 0seg. (281 kB/s)
Seleccionando el paquete gdisk previamente no seleccionado.
(Leyendo la base de datos ... 202241 ficheros o directorios instalados
actualmente.)
Preparing to unpack .../gdisk_0.8.8-1ubuntu0.1_amd64.deb ...
Unpacking gdisk (0.8.8-1ubuntu0.1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for doc-base (0.10.5) ...
Procesando 1 archivo doc-base añadido...
Configurando gdisk (0.8.8-1ubuntu0.1) ...
```

Sintaxis

```
gdisk <disco>
```

No se necesita ninguna opción. Abrimos el disco, grabamos y salimos:

```
# gdisk /dev/sdb
GPT fdisk (gdisk) version 0.8.8

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present

*****

Found invalid GPT and valid MBR; converting MBR to GPT format.

THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by typing 'q' if
```

```
you don't want to convert your MBR partitions to GPT format!
```

```
*****
```

```
Command (? for help):
```

Escribiendo w se solicita realizar la conversión:

```
Command (? for help): w
```

```
Final checks complete. About to write GPT data. THIS WILL  
OVERWRITE EXISTING  
PARTITIONS!!
```

```
Do you want to proceed? (Y/N):
```

Antes de que la operación se efectúe se solicita una confirmación. Responda y para validar o n para abandonar:

```
Do you want to proceed? (Y/N): y
```

```
OK; writing new GUID partition table (GPT) to /dev/sdb.
```

```
The operation has completed successfully.
```

d. Eliminar una tabla de particiones

Cuando la tabla de particiones se borra, los datos del disco se destruyen. Podemos eliminarla poniendo a cero el primer sector del disco con el comando dd.

Sintaxis

```
dd if=<source> of=<objetivo> bs=<tamaño_de_bloque> count=<valor>
```

Opciones necesarias:

Opción	Descripción
if=<fuente>	/dev/zero es un pseudo dispositivo que devuelve el carácter ASCII NUL, o sea 0x00. Nos permitirá borrar la información del sector 1.
of=<objetivo>	El objetivo es el disco correspondiente. Por ejemplo: /dev/sdb.
bs=<tamaño_del_bloque>	Define el tamaño del bloque.
count=<valor>	Copia una vez el bloque.

Ejemplo

Eliminación de la tabla de particiones del disco /dev/sdb:

```
# dd if=/dev/zero of=/dev/sdb bs=512 count=1
1+0 registros leídos
1+0 registros escritos
512 bytes (512 B) copiados, 0,00113457 s, 1,4 MB/s
```

2. Herramientas de particionado

Los programas de particionado que gestionan solo las particiones de tipo GPT son los siguientes:

- fdisk, una herramienta por línea de comandos.
- cfdisk, que es idéntica a fdisk a nivel de funcionalidad y posee una interfaz basada en la librería de software curses.

Las herramientas por línea de comandos que pueden gestionar los dos tipos de tablas de partición son las siguientes:

- GNU Parted, una herramienta por línea de comandos.
- gdisk, que es una evolución del comando fdisk.

a. fdisk

En todas las distribuciones de Linux, se proporciona fdisk. Crea, elimina y muestra las particiones.

He aquí la lista de particiones que soporta:

```
bob@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:~# fdisk /dev/sda  
  
Bienvenido a fdisk (util-linux 2.25.2).  
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.  
Tenga cuidado antes de utilizar la orden de escritura.  
  
Orden (n para obtener ayuda): l  
  
 0 Vacía                24 DOS de NEC          81 Minix / Linux a bf Solaris  
 1 FAT12                27 NTFS de WinRE o 82 Linux swap / So c1 DRDOS/sec (FAT-  
 2 XENIX root           39 Plan 9              83 Linux             c4 DRDOS/sec (FAT-  
 3 XENIX usr            3c PartitionMagic     84 Unidad C: ocult c6 DRDOS/sec (FAT-  
 4 FAT16 <32M          40 Venix 80286         85 Linux extendida c7 Syrix  
 5 Extendida           41 PPC PReP Boot      86 Conjunto de vol da Datos sin SF  
 6 FAT16               42 SFS                87 Conjunto de vol db CP/M / CTOS / .  
 7 HPFS/NTFS/exFAT     4d QNX4.x              88 Linux plaintext de Utilidad Dell  
 8 AIX                 4e QNX4.x segunda     8e Linux LVM        df BootIt  
 9 AIX arrancable      4f QNX4.x tercera     93 Amoeba           e1 DOS access  
 a Gestor de arran 50 OnTrack DM          94 Amoeba BBT       e3 DOS R/O  
 b W95 FAT32          51 OnTrack DM6 Aux  9f BSD/OS           e4 SpeedStor  
 c W95 FAT32 (LBA)    52 CP/M              a0 Hibernación de eb BeOS fs  
 e W95 FAT16 (LBA)    53 OnTrack DM6 Aux  a5 FreeBSD         ee GPT  
 f W95 Ext'd (LBA)    54 OnTrackDM6        a6 OpenBSD         ef EFI (FAT-12/16/  
10 OPUS              55 EZ-Drive          a7 NeXTSTEP        f0 inicio Linux/PA  
11 FAT12 oculta       56 Golden Bow       a8 UFS de Darwin   f1 SpeedStor  
12 Compaq diagnost 5c Priam Edisk       a9 NetBSD          f4 SpeedStor  
14 FAT16 oculta <3 61 SpeedStor    ab arranque de Dar f2 DOS secondary  
16 FAT16 oculta       63 GNU HURD o SysV af HFS / HFS+      fb VMFS de VMware  
17 HPFS/NTFS ocult 64 Novell Network b7 BSDI fs         fc VMKCORE de VMwa  
18 SmartSleep de A 65 Novell Network b8 BSDI swap       fd Linux raid auto  
1b FAT32 de W95 oc 70 DiskSecure Mult bb Boot Wizard hid fe LANstep  
1c FAT32 de W95 (L 75 PC/IX       be arranque de Sol ff BBT  
1e FAT16 de W95 (L 80 Minix antiguo  
  
Orden (n para obtener ayuda):
```

Las particiones utilizadas principalmente son Linux (83), swap (82) y LVM de Linux (8e).

Mostrar las particiones

En un terminal, escriba `fdisk -l` seguido del nombre de un disco como `/dev/sda`:

```
# fdisk -l /dev/sda
```

```
Disco /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectores
```

```
Unidades: sectores de 1 * 512 = 512 bytes
```

```
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
```

```
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
Tipo de etiqueta de disco: dos
```

```
Identificador del disco: 0x8c07f4df
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	104855551	104853504	50G	8e	Linux LVM

El disco /dev/sda contiene una sola partición de tipo 8e LVM de Linux: /dev/sda1.

Mostrar un disco sin tabla de partición

El disco /dev/sdb está «vacío»:

```
# fdisk -l /dev/sdb
```

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores
```

```
Unidades: sectores de 1 * 512 = 512 bytes
```

```
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
```

```
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
El disco /dev/sdb solo contiene una partición válida
```

Crear una tabla de partición

Ejecute fdisk /dev/sdb en un terminal:

```
# fdisk /dev/sdb
```

```
Bienvenido a fdisk (util-linux 2.25.2).
```

```
Los cambios solo permanecerán en la memoria, hasta que decida  
escribirlos.
```

```
Tenga cuidado antes de utilizar la orden de escritura.
```

```
El dispositivo no contiene una tabla de particiones reconocida.
```

```
Created a new DOS disklabel with disk identifier 0x1a81890b.
```

```
Orden (m para obtener ayuda):
```

Efectúe la siguiente operación:

Teclee m para mostrar la ayuda y encontrar la opción que permite crear una nueva tabla vacía de particiones DOS.

Use la opción o para crear una tabla de particiones.

Luego w para guardar y salir.

He aquí la ilustración de la operación:

```
Orden (m para obtener ayuda): m
```

```
Ayuda:
```

```
DOS (MBR)
```

- a conmuta el indicador de iniciable
- b modifica la etiqueta de disco BSD anidada
- c conmuta el indicador de compatibilidad con DOS

Generic

- d borra una partición
- l lista los tipos de particiones conocidos
- n añade una nueva partición
- p muestra la tabla de particiones
- t cambia el tipo de una partición
- v verifica la tabla de particiones

Miscelánea

- m muestra este menú
- u cambia las unidades de visualización/entrada
- x funciones adicionales (sólo para usuarios avanzados)

Guardar y Salir

- w escribe la tabla en el disco y sale
- q sale sin guardar los cambios

Crea una nueva etiqueta

- g crea una nueva tabla de particiones GPT vacía
- G crea una nueva tabla de particiones SGI (IRIX) vacía
- o crea una nueva tabla de particiones DOS vacía
- s crea una nueva tabla de particiones Sun vacía

Orden (m para obtener ayuda): o

Created a new DOS disklabel with disk identifier 0x215e65a3.

Orden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

Crear una nueva partición

El disco /dev/sdb posee una tabla de particiones de tipo msdos.

Ahora, ejecute fdisk /dev/sdb y responda a las preguntas:

Escriba n para crear una nueva partición y luego p para primaria.

Pulse la tecla [Intro] para confirmar el número de partición por defecto: 1.

Pulse la tecla [Intro] para confirmar el primer sector ofrecido por defecto: 2048.

Escriba +5G para el último sector.

Escriba p para ver su trabajo.

Escriba w para guardar y salir.

He aquí la ilustración de la operación:

```
# fdisk /dev/sdb
```

```
Orden (m para obtener ayuda): n
```

```
Tipo de partición
```

```
    p   primaria (0 primarias, 0 extendidas, 4 libres)
```

```
    e   extendida (contenedor para particiones lógicas)
```

```
Seleccionar (valor predeterminado p): p
```

```
Número de partición (1-4, valor predeterminado 1):
```

```
Primer sector (2048-20971519, valor predeterminado 2048):
```

```
Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-20971519,  
valor predeterminado 20971519): +5G
```

```
Crea una nueva partición 1 de tipo 'Linux' y de tamaño 5 GiB.
```

```
Orden (m para obtener ayuda): p
```

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores
```

```
Unidades: sectores de 1 * 512 = 512 bytes
```

```
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
```

```
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
Tipo de etiqueta de disco: dos
```

```
Identificador del disco: 0x5e303f1c
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	10487807	10485760	5G	83	Linux

Orden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

Se aprecia que el ID de la partición creada en /dev/sdb1 es 83 Linux.

Cree una segunda partición de 3 GB:

Escriba n para crear una nueva partición y luego p para primaria.

Pulse la tecla [Intro] para confirmar el número de partición por defecto: 2.

Pulse la tecla [Intro] para confirmar el primer sector ofrecido por defecto: 10487808.

Escriba +3G para el último sector.

Escriba p para ver su trabajo.

Escriba w para guardar y salir.

He aquí la ilustración de la operación:

```
# fdisk /dev/sdb
```

Orden (m para obtener ayuda): n

Tipo de partición

p primaria (1 primarias, 0 extendidas, 3 libres)

e extendida (contenedor para particiones lógicas)

Seleccionar (valor predeterminado p): p

Número de partición (2-4, valor predeterminado 2):

Primer sector (10487808-20971519, valor predeterminado 10487808):

Último sector, +sectores o +tamaño{K,M,G,T,P} (10487808-20971519, valor predeterminado 20971519): +3G

Crea una nueva partición 2 de tipo 'Linux' y de tamaño 3 GiB.

Orden (m para obtener ayuda): p

Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x5e303f1c

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	10487807	10485760	5G	83	Linux
/dev/sdb2		10487808	16779263	6291456	3G	83	Linux

Orden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

Eliminar una partición

El disco /dev/sdb tiene dos particiones primarias. Borre la segunda partición.

Ejecute `fdisk /dev/sdb` y realice las operaciones:

Escriba `d` para borrar una partición.

Escriba el número de la partición: 2.

Escriba `p` para ver el resultado.

Escriba `w` para guardar y salir.

He aquí la ilustración de la operación:

```
# fdisk /dev/sdb
```

Orden (m para obtener ayuda): d

Número de partición (1,2, valor predeterminado 2): 2v

Se ha borrado la partición 2.

Orden (m para obtener ayuda): p

Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x5e303f1c

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	10487807	10485760	5G	83	Linux

Orden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

Reparar el orden de los números de partición

Después de las operaciones de partición, a veces es posible que el orden de los números no sea correcto.

Ejecute `fdisk /dev/sdb` y realice la operación:

Pulse m para la ayuda.

Pulse x para ir a las funciones adicionales (usuarios avanzados).

Pulse m para la ayuda del menú avanzado.

Pulse f para corregir la orden de las particiones.

Pulse r para volver al menú principal y w para guardar y salir.

He aquí la ilustración de la operación:

```
# fdisk /dev/sdb
```

Orden (m para obtener ayuda): m

Ayuda:

DOS (MBR)

a conmuta el indicador de iniciable

- b modifica la etiqueta de disco BSD anidada
- c conmuta el indicador de compatibilidad con DOS

Generic

- d borra una partición
- l lista los tipos de particiones conocidos
- n añade una nueva partición
- p muestra la tabla de particiones
- t cambia el tipo de una partición
- v verifica la tabla de particiones

Miscelánea

- m muestra este menú
- u cambia las unidades de visualización/entrada
- x funciones adicionales (sólo para usuarios avanzados)

Guardar y Salir

- w escribe la tabla en el disco y sale
- q sale sin guardar los cambios

Crea una nueva etiqueta

- g crea una nueva tabla de particiones GPT vacía
- G crea una nueva tabla de particiones SGI (IRIX) vacía
- o crea una nueva tabla de particiones DOS vacía
- s crea una nueva tabla de particiones Sun vacía

Orden (m para obtener ayuda): x

Orden avanzada (m para obtener ayuda): m

Ayuda (operaciones de experto):

DOS (MBR)

- b mueve el principio de los datos de una partición
- i cambia el identificador de disco

Geometría

- c cambia el número de cilindros
- h cambia el número de cabezas
- s cambia el número de sectores por pista

Generic

- p muestra la tabla de particiones
- v verifica la tabla de particiones
- d muestra los datos en bruto del primer sector del dispositivo
- D imprime los datos en bruto de la etiqueta de disco del

dispositivo

- f corrige el orden de las particiones
- m muestra este menú

Guardar y Salir

- q sale sin guardar los cambios
- r vuelve al menú principal

Orden avanzada (m para obtener ayuda): f

No hay nada que hacer. El orden ya es correcto.

Orden avanzada (m para obtener ayuda): r

Orden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

b. cfdisk

Idéntico a fdisk, las operaciones son más sencillas de realizar.

Por defecto, si ejecuta cfdisk, se mostrará el primer disco:

```
bob@debiansrv: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
Disco: /dev/sda  
Tamaño: 8 GiB, 8589934592 bytes, 16777216 sectores  
Etiqueta: dos, identificador: 0x72544fbc  

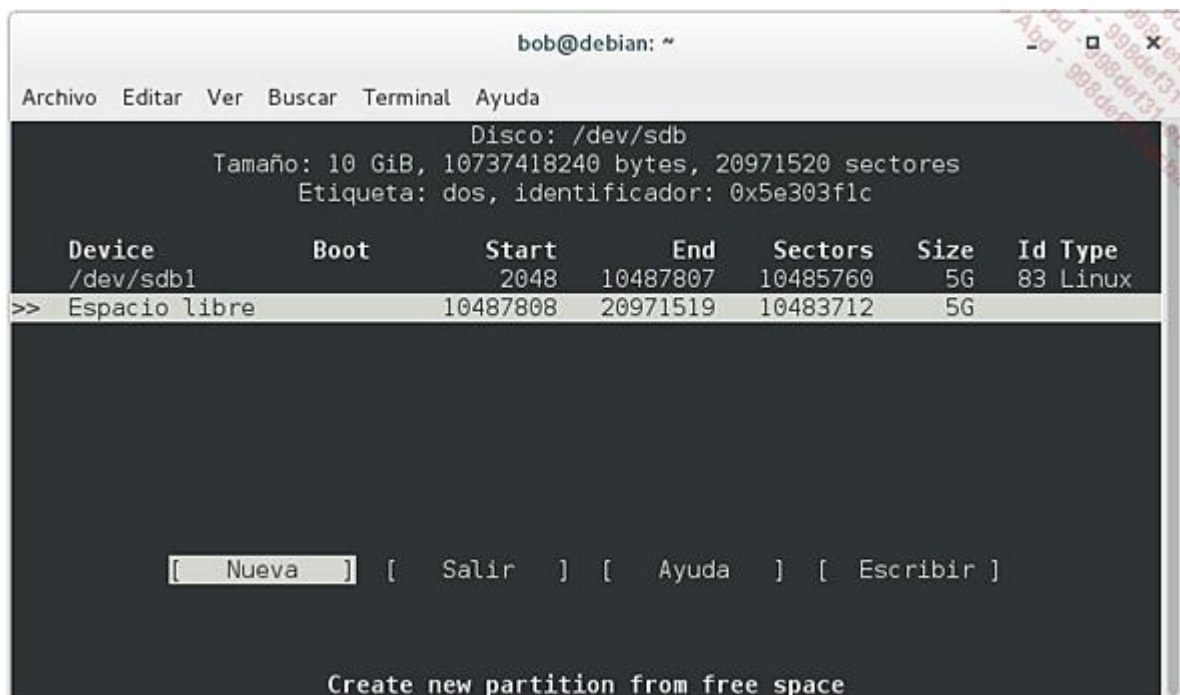

| Device       | Boot | Start   | End      | Sectors  | Size | Id | Type      |
|--------------|------|---------|----------|----------|------|----|-----------|
| >> /dev/sda1 | *    | 2048    | 1953791  | 1951744  | 953M | 83 | Linux     |
| /dev/sda2    |      | 1955838 | 16775167 | 14819330 | 7,1G | 5  | Extended  |
| └─/dev/sda5  |      | 1955840 | 16775167 | 14819328 | 7,1G | 8e | Linux LVM |

  
[ Inicializable ] [ Suprimir ] [ Salir ] [ Tipo ] [ Ayuda ]  
[ Escribir ]  
  
Toggle bootable flag of the current partition
```

Crear una nueva partición

Vamos a crear una partición primaria de 4 GB en el disco /dev/sdb.

Ejecutamos cfdisk /dev/sdb.



Y procedemos de la manera siguiente:

Ubicamos el cursor sobre el espacio libre de 5G.

Seleccionamos **Nueva**.

Seleccionamos **Primaria**.

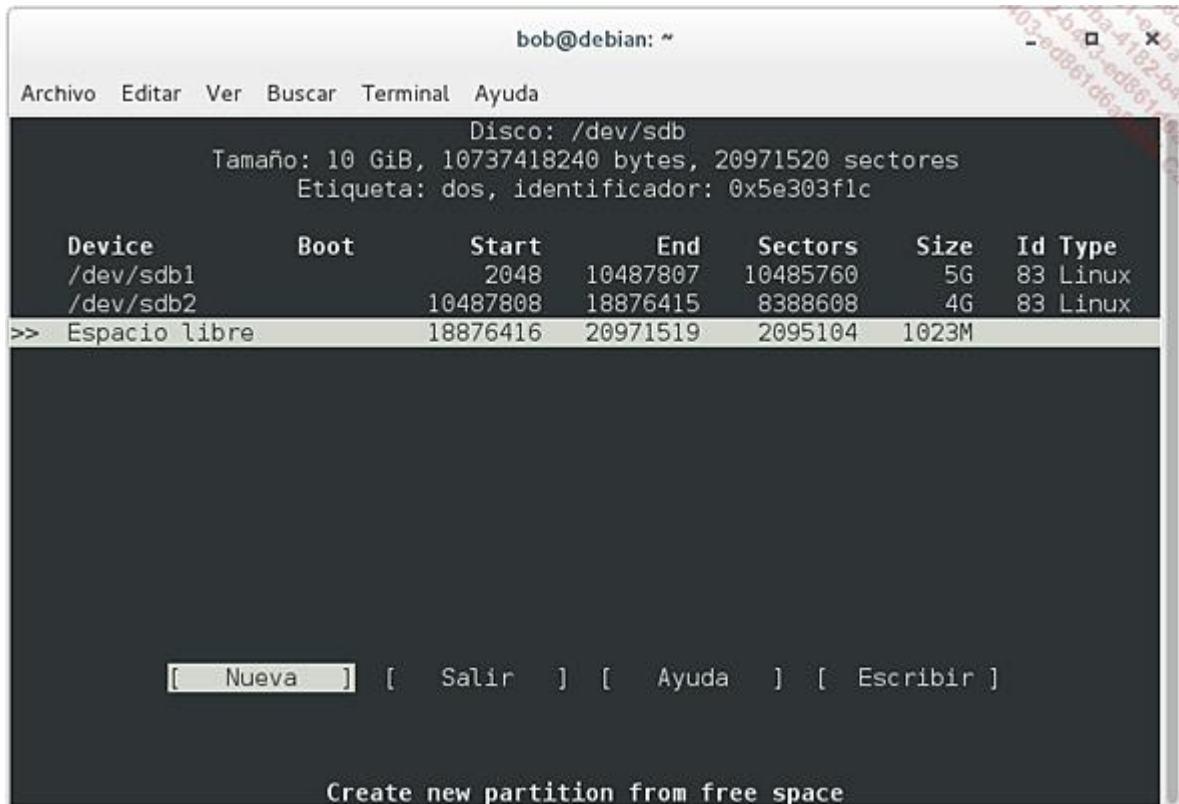
Introducimos el tamaño en GB, o sea 4G.

Seleccionamos **Iniciable**.

Seleccionamos **Escribir** para guardar.

Confirmamos con **Sí**.

He aquí el resultado de la operación:



```
bob@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

Disco: /dev/sdb
Tamaño: 10 GiB, 10737418240 bytes, 20971520 sectores
Etiqueta: dos, identificador: 0x5e303f1c

Device      Boot      Start      End  Sectors  Size  Id Type
/dev/sdb1                2048  10487807  10485760    5G   83 Linux
/dev/sdb2      10487808  18876415   8388608    4G   83 Linux
>> Espacio libre      18876416  20971519   2095104   1023M

[ Nueva ] [ Salir ] [ Ayuda ] [ Escribir ]

Create new partition from free space
```

Eliminar una partición

Vamos a eliminar /dev/sdb1:

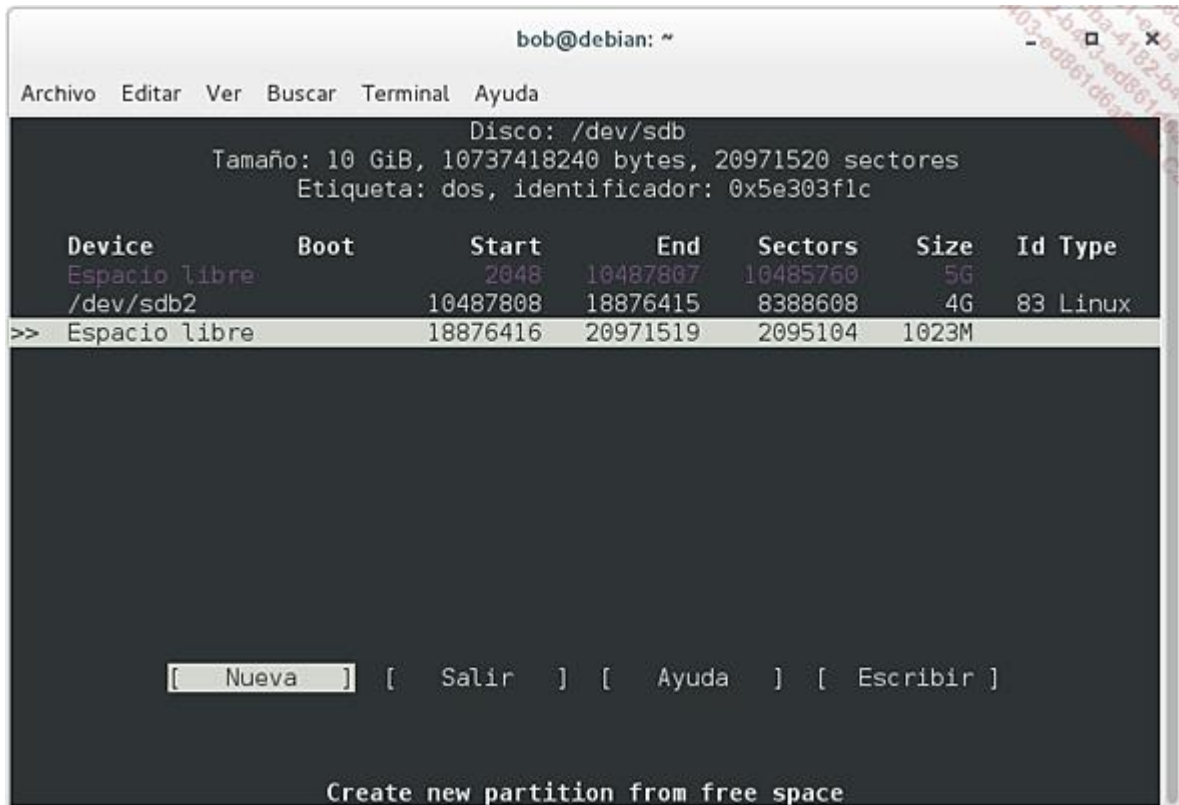
Coloque el cursor sobre la partición a eliminar, o sea la línea sdb1.

Seleccione **Suprimir**.

Seleccionamos **Escribir** para guardar.

Confirmamos con **Sí**.

He aquí el resultado de la operación:



```
bob@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Disco: /dev/sdb  
Tamaño: 10 GiB, 10737418240 bytes, 20971520 sectores  
Etiqueta: dos, identificador: 0x5e303f1c  


| Device           | Boot | Start    | End      | Sectors  | Size  | Id | Type  |
|------------------|------|----------|----------|----------|-------|----|-------|
| Espacio libre    |      | 2048     | 10487807 | 10485760 | 5G    |    |       |
| /dev/sdb2        |      | 10487808 | 18876415 | 8388608  | 4G    | 83 | Linux |
| >> Espacio libre |      | 18876416 | 20971519 | 2095104  | 1023M |    |       |

  
[ Nueva ] [ Salir ] [ Ayuda ] [ Escribir ]  
  
Create new partition from free space
```

c. GNU parted

Para gestionar las tablas de tipo msdos y GPT, utilizamos GNU parted.

La herramienta se encuentra por defecto en las distribuciones Ubuntu Server y CentOS. En cambio, habrá que instalarla en Debian.

Instalación Debian

Escriba lo siguiente:

```
# apt-get install parted
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma
automática y ya no son necesarios.
    apper apper-data brasero-cdrkit cdparanoia dvd+rw-tools dvdauthor
    fonts-sil-gentium fonts-sil-gentium-basic genisoimage
gir1.2-clutter-gst-2.0
    gir1.2-gtop-2.0 growisofs gvfs-libs hyphen-en-us k3b-data
kdeaccessibility
    kdesudo libatasmart4 libcdio-cdda1 libcdio-paranoia1
libdebconf-kde0
    libflac++6 libk3b6 libk3b6-extracodecs libkcddb4 libkcompactdisc4
    liblistaller-glib0 libpackagekitqt4-0 libqapt-runtime libqapt1
libreoffice
    libreoffice-help-en-us libreoffice-kde
libreoffice-report-builder-bin
    libudisks2-0 listaller mythes-en-us ntfs-3g python3-pykde4
python3-pyqt4
    python3-sip python3-software-properties qapt-batch
    software-properties-common software-properties-kde
unattended-upgrades
    vcdimager wodim
Utilice «apt-get autoremove» para eliminarlos.
Paquetes sugeridos:

    parted-doc
Se instalarán los siguientes paquetes NUEVOS:
    parted
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0
```

```
no actualizados.
Se necesita descargar 195 kB de archivos.
Se utilizarán 311 kB de espacio de disco adicional después de esta
operación.
Des:1 http://ftp.es.debian.org/debian/ jessie/main parted amd64
3.2-7 [195 kB]
Descargados 195 kB en 0s (2.401 kB/s)
Seleccionando el paquete parted previamente no seleccionado.
(Leyendo la base de datos ... 250476 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../parted_3.2-7_amd64.deb ...
Desempaquetando parted (3.2-7) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Configurando parted (3.2-7) ...
```

Crear una tabla de partición de tipo GPT

La ejecución en un terminal del software GNU parted mostrará un prompt (parted). Siga este procedimiento:

Escriba `select /dev/sdb` para seleccionar el disco.

Luego escriba `print` para ver el estado del disco.

He aquí la ilustración de la operación:

```
# parted
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/sdb
Using /dev/sdb
(parted) print

Error: /dev/sdb: unrecognised disk label
(parted)
```

El disco está vacío. Debemos crear una tabla de partición.

Escriba `mklabel gpt` para crear la tabla de partición de tipo GPT.

Luego escriba print de nuevo para mostrar el estado del disco.

He aquí la ilustración de la operación:

```
(parted) mklabel gpt
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

(parted)
```

Crear una partición

Para crear una partición de 4 GB llamada «data»:

Escriba mkpart data 1 4000.

Luego escriba print para mostrar el estado del disco.

He aquí la ilustración de la operación:

```
(parted) mkpart data 1 4000
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

1       1049kB  4000MB  3999MB                data

(parted)
```

Eliminar una partición

Para efectuar la eliminación de la partición 1:

Escriba `rm 1`.

Luego escriba `print` para ver el estado del disco.

Salga escribiendo la letra `q`.

Aquí vemos la operación:

```
(parted) mklabel gpt
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags

```

(parted) `q`

Information: You may need to update `/etc/fstab`.

d. GParted y QtParted

Si contamos con un entorno gráfico como GNOME 3, podemos utilizar GParted, un front-end de GNU parted escrito en C++ y GTK+.

Destinado a KDE, QtParted es también un front-end de GNU parted escrito en C++ y Qt.

Instalación Debian

Cómo instalar GParted:

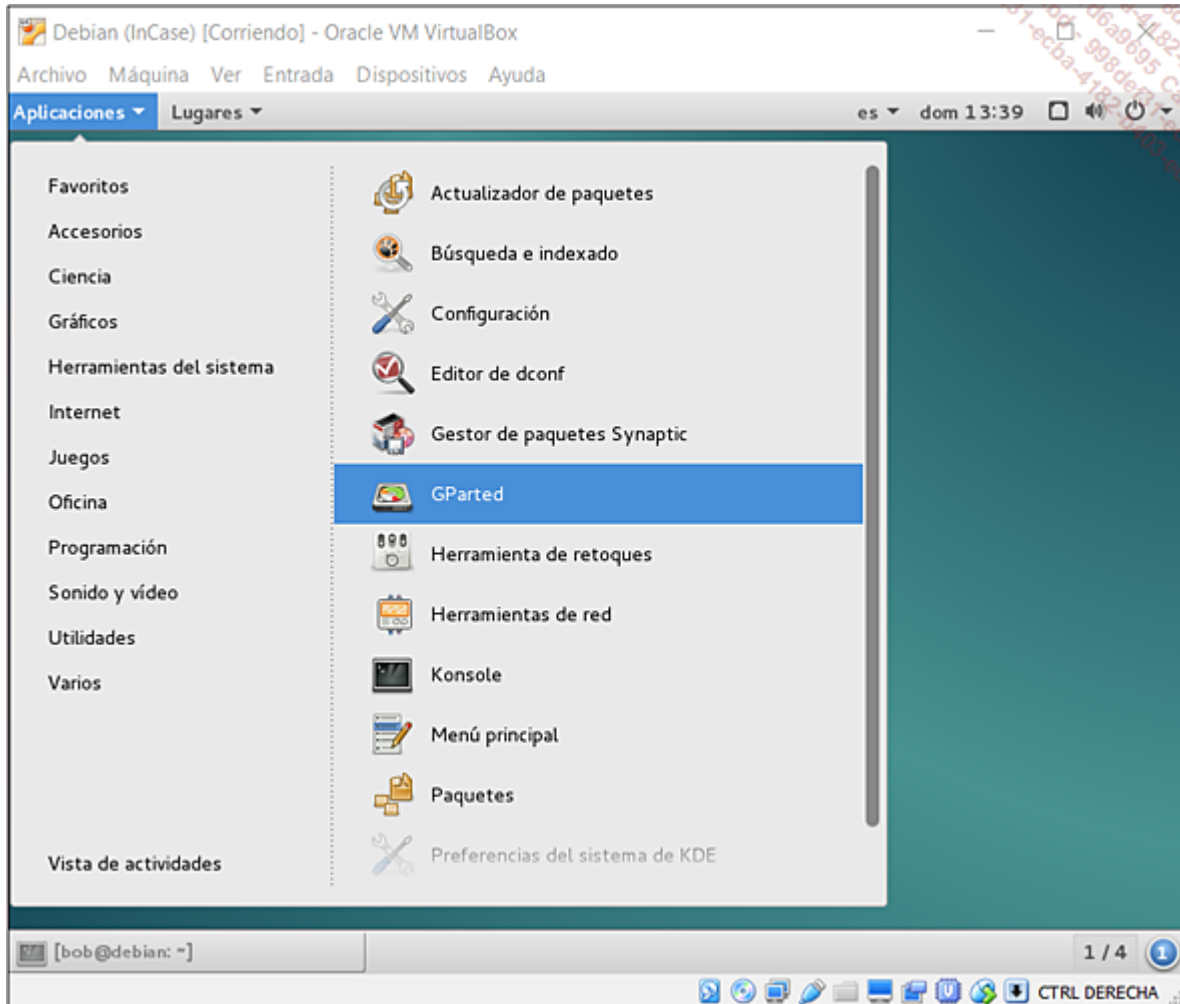
```
# apt-get -y install gparted
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libparted-fs-resize0
Paquetes sugeridos:
  xfsprogs reiserfsprogs reiser4progs jfsutils ntfs-3g mtools
kpartx dmraid gpart libparted-dev
Se instalarán los siguientes paquetes NUEVOS:
  gparted libparted-fs-resize0
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 2.127 kB de archivos.
Se utilizarán 7.055 kB de espacio de disco adicional después de
esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ftp.es.debian.org/debian/ jessie/main
libparted-fs-resize0 amd64 3.2-7 [200 kB]
Des:2 http://ftp.es.debian.org/debian/ jessie/main gparted
amd64 0.19.0-2 [1.927 kB]
Descargados 2.127 kB en 5s (377 kB/s)
Seleccionando el paquete libparted-fs-resize0:amd64 previamente
no seleccionado.
(Leyendo la base de datos ... 249199 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar
.../libparted-fs-resize0_3.2-7_amd64.deb ...
Desempaquetando libparted-fs-resize0:amd64 (3.2-7) ...
Seleccionando el paquete gparted previamente no seleccionado.
Preparando para desempaquetar .../gparted_0.19.0-2_amd64.deb ...
Desempaquetando gparted (0.19.0-2) ...
```

*** Salida truncada ***

Sinopsis

En el menú de GNOME 3:

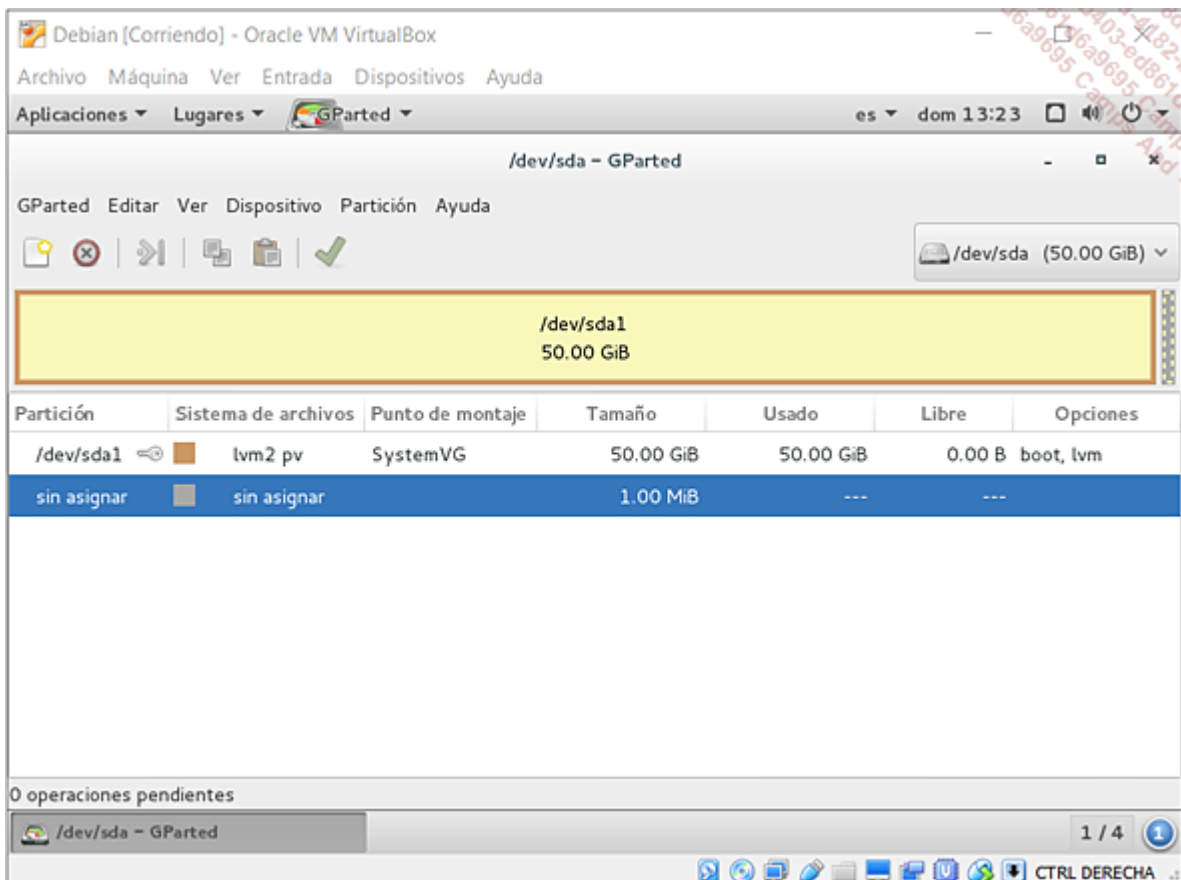
Aplicaciones - Herramientas del sistema - GParted como vemos en la siguiente imagen:



Se requiere la autenticación para tener privilegios de root:



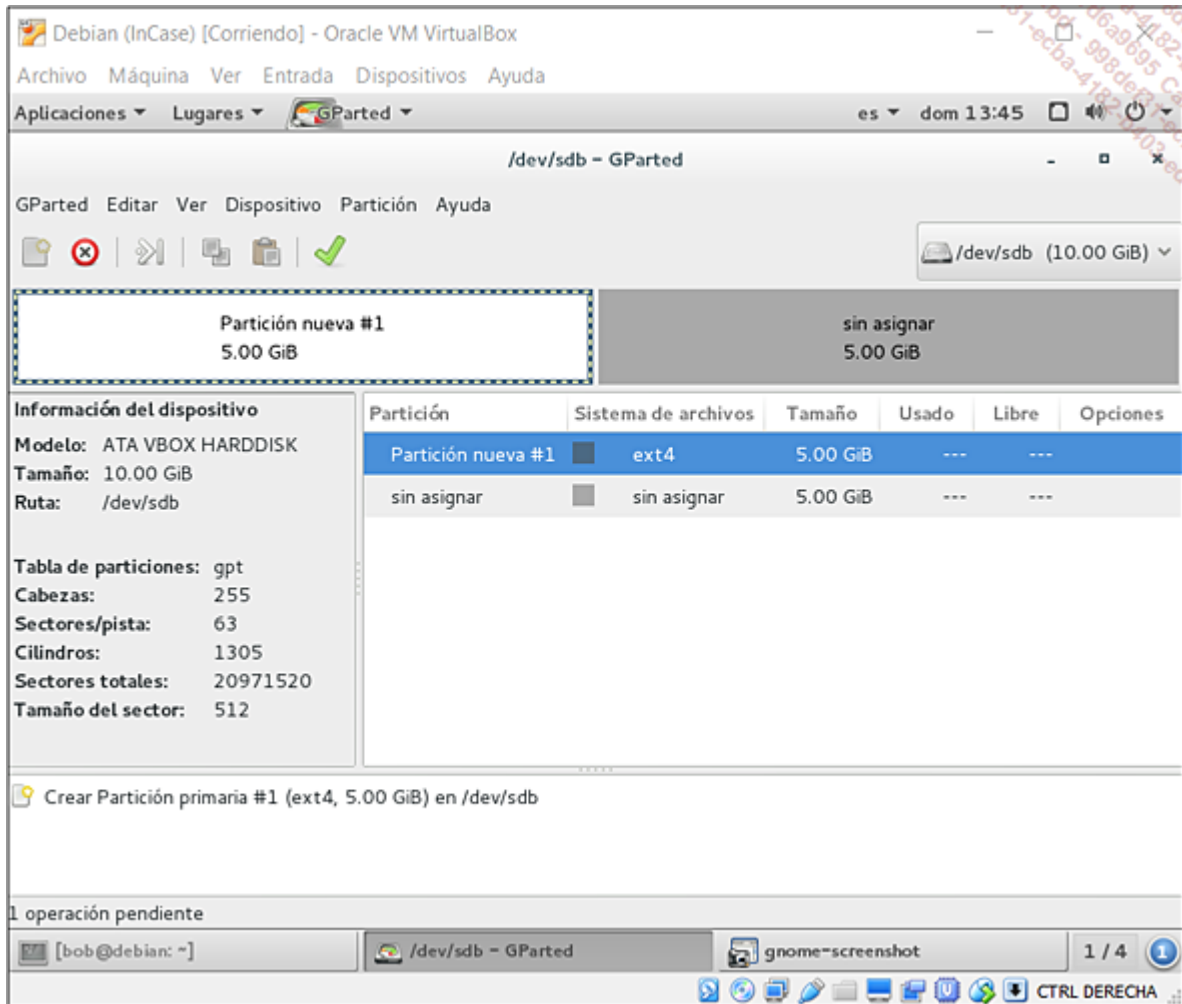
En cuanto se valide la contraseña de su usuario, GParted muestra su ventana principal:



Haga clic en el menú **Ver - Información del dispositivo**.

Seleccionamos en la parte superior derecha el disco /dev/sdb.

Aquí vemos la operación:



Vemos la partición de 5 GiB que hemos creado en el ejemplo anterior. Un «#1» en la columna **Partición** significa que esta no se ha formateado todavía y que por tanto no contiene ningún sistema de archivos.

e. Live CD

El uso de una distribución Linux «Live» en CD o llave USB también puede ser una solución para la gestión de particiones:

- GParted,
- Knoppix,
- Puppy Linux...

Logical Volume Manager

La gestión de volúmenes lógicos permite una gestión de más alto nivel de los problemas de almacenamiento en disco que el enfoque tradicional con discos y particiones.

El administrador del sistema tiene una mayor flexibilidad para asignar el espacio a las aplicaciones y a los usuarios.

Existen dos versiones de LVM:

- LVM versión 1 que se encuentra en los kernels 2.4.
- LVM versión 2, que es casi totalmente compatible con los volúmenes creados con LVM1, con excepción del soporte de instantáneas y de clúster. Podemos convertir un grupo de volúmenes del formato LVM1 al formato LVM2 con el comando `vgconvert`.

1. Terminología

a. Volumen físico

El volumen físico (PV o physical volume) es un disco o partición como `/dev/sdb1` que se divide en segmentos de 4 MB llamados extensiones físicas (PE o physical extents).

PV 1

PE de 4MB
PE de 4MB
PE de 4MB
PE de 4MB
PE de 4MB
PE de 4MB
...

El tamaño de un PE es por defecto de 4 MB. Su valor puede modificarse según las necesidades.

b. Grupo de volúmenes

Un grupo de volúmenes (VG o volume group) es un conjunto de PV. Un PV no puede pertenecer a varios VG.

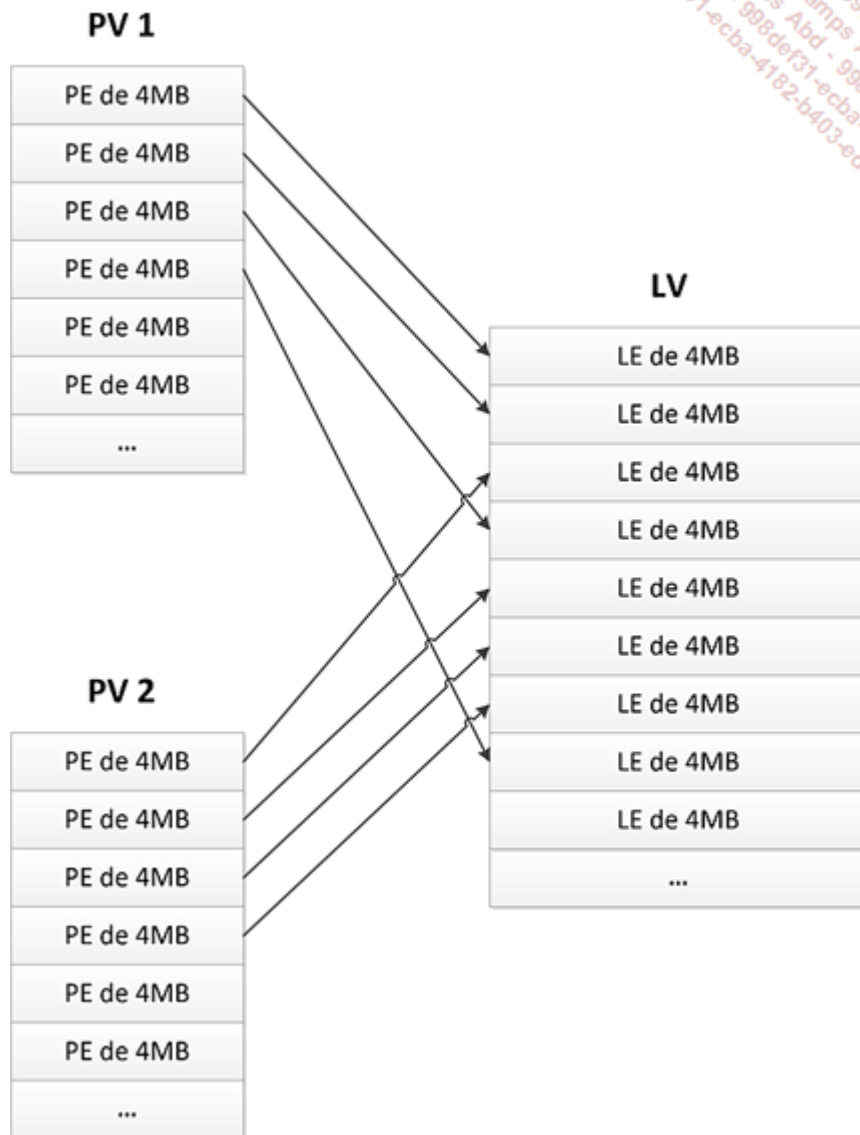
c. Volumen Lógico

Un volumen lógico (LV o logical volume) es un espacio de almacenamiento que posee, en consecuencia, un sistema de archivos. Un LV está dividido en segmentos de 4 MB llamados extensiones lógicas (LE o logical extents):

LV

LE de 4MB
LE de 4MB
LE de 4MB
LE de 4MB
LE de 4MB
LE de 4MB
LE de 4MB
LE de 4MB
...

LVM crea punteros que relacionan los PE con los LE:



d. Metadatos

LVM utiliza metadatos (metadatos) para administrar el sistema.

- La Physical Volume Reserved Area (PVRA) que contiene información sobre los volúmenes físicos creados por LVM.
- El Volume Group Reserved Area (VGRA) que contiene información sobre los grupos de volúmenes y volúmenes lógicos incluidos en estos grupos.
- El Bad Block Relocation Area (BBRA) es una zona que contiene la información y su enlace a un mecanismo de redistribución de los bloques dañados, permitiendo así la recuperación automática de bloques defectuosos.

2. Administración de volúmenes físicos

Los pedidos que gestionan los volúmenes físicos empiezan por pv y se almacenan en la carpeta /sbin:

```
# ls /sbin/pv*
/sbin/pvchange  /sbin/pvcreate  /sbin/pvmove
/sbin/pvresize  /sbin/pvscan    /sbin/pvck
/sbin/pvdisplay /sbin/pvremove  /sbin/pvs
```

a. Crear un volumen físico

Hay que inicializar el disco o partición con pvcreate para crear un PV. Antes de actuar, podemos hacer un balance de los discos con lvmdiskscan.

Sintaxis

```
pvcreate <discos_o_particiones>
```

Puede inicializar uno o varios dispositivos de bloque al mismo tiempo. Basta con nombrarlos separados por un espacio.

Sintaxis

```
lvmdiskscan
```

Este comando no tiene opciones.

Ejemplo con Debian

Mostrar los discos del sistema:

```
# lvmdiskscan
/dev/root          [      13,97 GiB]
/dev/sda1          [      50,00 GiB] LVM physical volume
/dev/dm-1          [      488,00 MiB]
/dev/SystemVG/homeLV [      35,39 GiB]
/dev/SystemVG/optLV  [      168,00 MiB]
/dev/sdb1          [       3,72 GiB]
/dev/sdc           [      10,00 GiB]
/dev/sdd           [      10,00 GiB]
/dev/sde           [      10,00 GiB]
```

```
6 disks
2 partitions
0 LVM physical volume whole disks
1 LVM physical volume
```

El objetivo es la inicialización de dos discos: /dev/sdb y /dev/sdc.

El disco /dev/sdc está «vacío», al igual que /dev/sdd y /dev/sde. Sin embargo, ya existe una tabla de particiones en el disco /dev/sdb y contiene una partición /dev/sdb1. Si ejecutamos pvcreate se producirá un error:

```
# pvcreate /dev/sdb /dev/sdc
Device /dev/sdb not found (or ignored by filtering).
Writing physical volume data to disk "/dev/sdc"
Physical volume "/dev/sdc" successfully created
```

Antes de proceder, debemos borrar con parted la partición /dev/sdb1:

```
# parted /dev/sdb
GNU Parted 2.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) rm 1
(parted) q
```

Elimine la tabla de particiones del disco /dev/sdb:

```
# dd if=/dev/zero of=/dev/sdb bs=512 count=1
1+0 registros leídos
1+0 registros escritos
512 bytes (512 B) copiados, 0,0133707 s, 1,4 MB/s
```

Ahora podemos llevar a cabo la creación de las dos PV:

```
# pvcreate /dev/sdb /dev/sdc
Writing physical volume data to disk "/dev/sdb"
Physical volume "/dev/sdb" successfully created
Writing physical volume data to disk "/dev/sdc"
Physical volume "/dev/sdc" successfully created
```

Verificamos:

```
# lvm diskscan
/dev/root          [      13,97 GiB]
/dev/sda1          [      50,00 GiB] LVM physical volume
/dev/dm-1          [      488,00 MiB]
/dev/SystemVG/homeLV [      35,39 GiB]
/dev/SystemVG/optLV  [      168,00 MiB]
/dev/sdb           [      10,00 GiB] LVM physical volume
/dev/sdc           [      10,00 GiB] LVM physical volume
/dev/sdd           [      10,00 GiB]
/dev/sde           [      10,00 GiB]

5 disks
1 partition
2 LVM physical volume whole disks
1 LVM physical volume
```

b. Mostrar los volúmenes físicos

Disponemos de tres comandos para mostrar los PV existentes:

- `pvs` devuelve la información de los PV,
- `pvscan` enumera los PV,
- `pvdisplay` muestra las propiedades de un PV.

Sintaxis

```
pvs
```

No se requieren opciones para `pvs`.

Sintaxis

```
pvscan <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-s	--short	Vista rápida de los PV.
-u	--uuid	Añade a la pantalla el UUID (Uniform Unique Identifiers).

Sintaxis

```
pvdisplay <opción>
```

Opción útil:

Opción		Descripción
corta	larga	
-s	--short	Muestra solo el tamaño del PV.

Ejemplos con Debian

Podemos ver el resultado con pvs después de la inicialización de los dos discos /dev/sdb y /dev/sdc.

```
# pvs
PV          VG          Fmt  Attr  Psize  Pfree
/dev/sda1   SystemVG   lvm2  a--   50,00g  0
/dev/sdb           lvm2  a--   10,00g  10,00g
/dev/sdc           lvm2  a--   10,00g  10,00g
```

Con pvscan:

```
# pvscan -u
PV /dev/sda1 with UUID OyaFrT-Hoa8-1L8z-fnTY-eyJt-Utf7-Hs3aw2 VG
SystemVG          lvm2 [50,00 GiB / 0    free]
PV /dev/sdb  with UUID xG0az2-9zw8-C5je-ThHR-AFqQ-fd0j-qb9h8K
```

```
lvm2 [10,00 GiB]
  PV /dev/sdc with UUID vZcG9T-Tjm1-907Y-7ZRo-qcUz-BmYF-Trurpe
lvm2 [10,00 GiB]
  Total: 3 [70,00 GiB] / in use: 1 [50,00 GiB] / in no VG: 2 [20,00 GiB]
```

Con pvdisplay:

```
# pvdisplay -s
Device "/dev/sda1" has a capacity of 0
Device "/dev/sdb" has a capacity of 10,00 GiB
Device "/dev/sdc" has a capacity of 10,00 GiB
```

O bien de forma más detallada:

```
# pvdisplay
--- Physical volume ---
PV Name                /dev/sda1
VG Name                SystemVG
PV Size                50,00 GiB / not usable 2,00 MiB
Allocatable            yes (but full)
PE Size                4,00 MiB
Total PE               12799
Free PE                0
Allocated PE           12799
PV UUID                OyaFrT-Hoa8-1L8z-fnTY-eyJt-Utf7-Hs3aw2

"/dev/sdb" is a new physical volume of "10,00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb
VG Name
PV Size                10,00 GiB
Allocatable            NO
PE Size                0
Total PE               0

Free PE                0
```

```

Allocated PE          0
PV UUID               xG0az2-9zw8-C5je-ThHR-AFqQ-fd0j-qb9h8K

"/dev/sdc" is a new physical volume of "10,00 GiB"
--- NEW Physical volume ---
PV Name               /dev/sdc
VG Name
PV Size               10,00 GiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               vZcG9T-Tjm1-907Y-7ZRo-qcUz-BmYF-Trurpe

```

c. Cambiar el tamaño de un volumen físico

Sintaxis

```
pvresize <opción> <pv>
```

Opción útil:

Opción larga	Descripción
<code>--setphysicalvolumesize <valor></code>	Define el nuevo tamaño del PV según <valor> para el PV pasado como argumento.

Ejemplo con Debian

Mostrar los PV:

```

# pvscan
PV /dev/sda1   VG SystemVG      lvm2 [50,00 GiB / 0   free]
PV /dev/sdb    VG SystemVG      lvm2 [10,00 GiB]
PV /dev/sdc    VG SystemVG      lvm2 [10,00 GiB]

Total: 3 [70,00 GiB] / in use: 1 [50,00 GiB] / in no VG: 2

```

```
[20,00 GiB]
```

Reducir /dev/sdb en 2 GB:

```
# pvresize --setphysicalvolumesize 8G /dev/sdb
Physical volume "/dev/sdb" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

Verificamos:

```
# pvscan
PV /dev/sda1   VG SystemVG      lvm2 [50,00 GiB / 0   free]
PV /dev/sdb                lvm2 [8,00 GiB]
PV /dev/sdc                lvm2 [10,00 GiB]
Total: 3 [68,00 GiB] / in use: 1 [50,00 GiB] / in no VG: 2
[18,00 GiB]
```

d. Eliminar un volumen físico

La eliminación de un PV que no se ha integrado en un VG, se efectúa con el comando `pvremove`.

Sintaxis

```
pvremove <pv>
```

<pv> es la ruta del dispositivo de bloque como /dev/sda2.

Ejemplo con Debian

Mostrar los PV:

```
# pvs
PV          VG          Fmt  Attr  Psize  Pfree
/dev/sda1   SystemVG   lvm2  a--   50,00g    0
/dev/sdb                lvm2  a--    8,00g    8,00g
/dev/sdc                lvm2  a--   10,00g   10,00g
```

Eliminar el PV /dev/sdc:

```
# pvremove /dev/sdc
Labels on physical volume "/dev/sdc" successfully wiped
```


Verificamos:

```
# pvs
  PV      VG      Fmt      Attr      Psize      Pfree
  /dev/sda1  SystemVG      lvm2      a--      50,00g      0
  /dev/sdb           lvm2      a--      8,00g      8,00g
```

3. Administración de grupos de volúmenes

Los comandos que gestionan los grupos de volúmenes empiezan por vg y se almacenan en la carpeta /sbin:

```
# ls /sbin/vg*
/sbin/vgcfgbackup  /sbin/vgck          /sbin/vgdisplay
/sbin/vgimport     /sbin/vgmknodes     /sbin/vgrename
/sbin/vgsplit      /sbin/vgcfgrestore  /sbin/vgconvert
/sbin/vgexport     /sbin/vgimportclone /sbin/vgreduce
/sbin/vgs          /sbin/vgchange       /sbin/vgcreate
/sbin/vgextend     /sbin/vgmerge       /sbin/vgremove
/sbin/vgscan
```

a. Crear un grupo de volúmenes

La creación de un VG se realiza con el comando vgcreate.

Sintaxis

```
vgcreate <opciones> <nombre_de_VG> <lista_PV>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-l <valor>	--maxlogicalvolumes <valor>	Establece el número máximo de LV en el VG.
-p <valor>	--maxphysicalvolumes <valor>	Establece el número máximo de PV en el VG.

Ejemplo con Debian

Crear un VG que se llame dataVG y añadir dos PV /dev/sdb y /dev/sdc:

```
# vgcreate dataVG /dev/sdb /dev/sdc
Volume group "dataVG" successfully created
```

b. Mostrar los grupos de volúmenes

Disponemos de tres comandos para mostrar los VG existentes:

- vgs devuelve la información de los VG.
- vgscan enumera los VG y actualiza la caché.
- vgdisplay muestra las propiedades de un VG.

Sintaxis

```
vgs
```

No se requieren opciones para vgs.

Sintaxis

```
vgscan <opciones>
```

No se requieren opciones para vgscan.

Sintaxis

```
vgdisplay <opciones> <nombre_de_VG>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-a	--activevolumegroups	Muestra los VG activos.
-s	--short	Vista rápida de los VG.
-v	--verbose	Visualización detallada del VG.
-vv	--verbose --verbose	Visualización más detallada del VG.
-vvv	--verbose --verbose --verbose	Visualización todavía más detallada del VG.

Si se omite el nombre del VG, vgdisplay mostrará todos los VG.

Ejemplo con Debian

Mostrar los VG:

```
# vgs
```

VG	#PV	#LV	#SN	Attr	Vsize	VFree
SystemVG	1	4	0	wz-n-	50,00g	0
dataVG	2	0	0	wz-n-	17,99g	17,99g

Comprobamos que el VG dataVG contiene 2 PV, 0 LV y dispone de unos 20 GB en torno a GB.

Con vgscan:

```
# vgscan
```

```
Reading all physical volumes. This may take a while...
Found volume group "dataVG" using metadata type lvm2
Found volume group "SystemVG" using metadata type lvm2
```

Con vgdisplay:

```
# vgdisplay dataVG
--- Volume group ---
VG Name                dataVG
System ID
Format                 lvm2
Metadata Areas         2
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                 2
Act PV                  2
VG Size                 17,99 GiB
PE Size                 4,00 MiB
Total PE                4606
Alloc PE / Size         0 / 0
Free PE / Size          4606 / 17,99 GiB
VG UUID                 WtJTkp-k8Pj-Kphw-UTwp-CsFq-FmWI-VGz1Ce
```

O bien de forma más detallada:

```
# vgdisplay -v dataVG
Using volume group(s) on command line
Finding volume group "dataVG"
--- Volume group ---
VG Name                dataVG
System ID
Format                 lvm2
Metadata Areas         2

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           2
Act PV            2
VG Size           17,99 GiB
PE Size           4,00 MiB
Total PE          4606
Alloc PE / Size   0 / 0
Free PE / Size    4606 / 17,99 GiB
VG UUID           WtJTkp-k8Pj-Kphw-UTwp-CsFq-FmWI-VGz1Ce
```

--- Physical volumes ---

```
PV Name           /dev/sdb
PV UUID           rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status         allocatable
Total PE / Free PE 2047 / 2047
```

```
PV Name           /dev/sdc
PV UUID           m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status         allocatable
Total PE / Free PE 2559 / 2559
```

c. Añadir un volumen físico al grupo de volúmenes

Para añadir un PV y al VG usamos el comando `vgextend`.

Sintaxis

```
vgextend <nombre_de_VG> <PV_a_añadir>
```

Este comando no tiene opciones.

Ejemplo con Debian

Con `lvmdiskscan`, consulte los discos disponibles:

```
# lvm diskscan
/dev/root          [      13,97 GiB]
/dev/sda1          [      50,00 GiB] LVM physical volume
/dev/dm-1          [      488,00 MiB]
/dev/SystemVG/homeLV [      35,39 GiB]
/dev/SystemVG/optLV  [      168,00 MiB]
/dev/sdb           [      10,00 GiB] LVM physical volume
/dev/sdc           [      10,00 GiB] LVM physical volume
/dev/sdd           [      10,00 GiB]
/dev/sde           [      10,00 GiB]
5 disks
1 partition
2 LVM physical volume whole disks
1 LVM physical volume
```

/dev/sdd y /dev/sde están disponibles. Vamos a añadir dev/sdd.

Debemos en primer lugar inicializar /dev/sdd como PV:

```
# pvcreate /dev/sdd
Writing physical volume data to disk "/dev/sdd"
Physical volume "/dev/sdd" successfully created
```

Añadir este PV al VG dataVG:

```
# vgextend dataVG /dev/sdd
Volume group "dataVG" successfully extended
```

Verificamos:

```
# vgs
Using volume group(s) on command line
Finding volume group "dataVG"
--- Volume group ---
VG Name                dataVG

System ID

Format                 lvm2
```

Metadata Areas	3
Metadata Sequence No	2
VG Access	read/write
VG Status	resizable
MAX LV	0
Cur LV	0
Open LV	0
Max PV	0
Cur PV	3
Act PV	3
VG Size	27,99 GiB
PE Size	4,00 MiB
Total PE	7165
Alloc PE / Size	0 / 0
Free PE / Size	7165 / 27,99 GiB
VG UUID	WtJTkP-k8Pj-Kphw-UTwp-CsFq-FmWI-VGz1Ce

--- Physical volumes ---

PV Name	/dev/sdb
PV UUID	rDOPej-jvBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status	allocatable
Total PE / Free PE	2047 / 2047

PV Name	/dev/sdc
PV UUID	m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status	allocatable
Total PE / Free PE	2559 / 2559

PV Name	/dev/sdd
PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
PV Status	allocatable
Total PE / Free PE	2559 / 2559

d. Retirar un volumen físico del grupo de volúmenes

Nos aseguramos de que el PV no está siendo utilizado con `vgdisplay`. Si este fuera el caso, usamos el comando `vgreduce`.

Si, por el contrario, el PV está siendo utilizado, será necesario recurrir al comando `pvremove` para migrar los datos a otro PV. Una vez realizada la operación, podemos eliminar el PV con `vgreduce`.

Sintaxis

```
vgreduce <nombre_del_VG> <PV_a_eliminar>
```

Este comando no tiene opciones.

Ejemplo con Debian

El objetivo es retirar `/dev/sdd` del PV `dataVG`.

Comprobamos si el PV `/dev/sdd` está utilizado en el VG:

```
# pvdisplay /dev/sdd
--- Physical volume ---
PV Name                /dev/sdd
VG Name                dataVG
PV Size                10,00 GiB / not usable 4,00 MiB
Allocatable            yes
PE Size                4,00 MiB
Total PE               2559
Free PE                2559
Allocated PE           0
PV UUID                msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
```

Free PE es igual a Total PE y Allocated PE vale 0. Por lo tanto, el PV no se está utilizando.

Retire el PV del VG `dataVG`:

```
# vgreduce dataVG /dev/sdd
Removed "/dev/sdd" from volume group "dataVG"
```

Verificamos:

```
# pvdisplay /dev/sdd
"/dev/sdd" is a new physical volume of "10,00 GiB"
--- NEW Physical volume ---
```


PV Name	/dev/sdd
VG Name	
PV Size	10,00 GiB
Allocatable	NO
PE Size	0
Total PE	0
Free PE	0
Allocated PE	0
PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7

El campo VG Name está vacío.

e. Activar o desactivar un grupo de volúmenes

El comando `vgchange` gestiona la disponibilidad de los LV del VG para las E/S.

Sintaxis

```
vgchange <opción> <nombre_de_VG>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-a <valor>	--available <valor>	Si <valor> es n entonces el VG se encuentra desactivado, de lo contrario, y para activar.

Si el VG no tiene LV y solicitamos la desactivación:

```
# vgchange -a n dataVG
```

```
0 logical volume(s) in volume group "dataVG" now active
```

El VG estará necesariamente inaccesible de todas formas.

f. Eliminar un grupo de volúmenes

Para la eliminación de un VG que no contiene LV, usamos el comando `vgremove`.

Sintaxis

```
vgremove <nombre_de_VG>
```

Este comando no tiene opciones.

Ejemplo con Debian

El objetivo es eliminar el VG `dataVG`.

```
# vgrremove dataVG
Volume group "dataVG" successfully removed
```

Verificamos:

```
# vgdisplay -s
"SystemVG" 50,00 GiB [50,00 GiB used / 0 free]
```

g. Dividir un grupo de volúmenes

La división de un VG se efectúa con el comando `vgsplit`.

Sintaxis

```
vgsplit <nombre_de_VG>
```

Este comando no tiene opciones.

Ejemplo con Debian

En primer lugar, debemos crear un VG con PV para poder realizar un ejemplo. Creamos un VG `appVG` con los PV `/dev/sdb`, `/dev/sdc` y `/dev/sdd`:

```
# vgcreate appVG /dev/sdb /dev/sdc /dev/sdd
Volume group "appVG" successfully created
```

El objetivo es dividir el VG `AppVG` de la siguiente manera:

- `AppVG` incluye los dos PV `/dev/sdb` y `/dev/sdc`,
- Un nuevo VG llamado `devVG` debe contener `/dev/sdd`.

Teclee en un terminal:

```
# vgsplit appVG devVG /dev/sdd  
New volume group "devVG" successfully split from "appVG"
```

Verificamos:

```
# vgdisplay -v appVG devVG  
Using volume group(s) on command line  
Finding volume group "appVG"  
--- Volume group ---  
VG Name                appVG  
System ID  
Format                 lvm2  
Metadata Areas         2  
Metadata Sequence No   2  
VG Access              read/write  
VG Status              resizable  
MAX LV                 0  
Cur LV                0  
Open LV               0  
Max PV                 0  
Cur PV                2  
Act PV                2  
VG Size                19,99 GiB  
PE Size                4,00 MiB  
Total PE              5118  
Alloc PE / Size        0 / 0  
Free PE / Size         5118 / 19,99 GiB  
VG UUID                xdeSxK-e2Lu-YKU1-bTMi-XN1K-mJkA-p2A6Fm  
  
--- Physical volumes ---  
  
PV Name                /dev/sdb  
PV UUID                rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
```

PV Status allocatable
Total PE / Free PE 2559 / 2559

PV Name /dev/sdc
PV UUID m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status allocatable
Total PE / Free PE 2559 / 2559

 Finding volume group "devVG"
--- Volume group ---

VG Name devVG
System ID
Format lvm2
Metadata Areas 1
Metadata Sequence No 2
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 GiB
PE Size 4,00 MiB
Total PE 2559
Alloc PE / Size 0 / 0
Free PE / Size 2559 / 10,00 GiB
VG UUID 0thp9m-LqH0-7lH0-o74o-YSNC-s5IS-BhBbSo

--- Physical volumes ---

PV Name /dev/sdd

PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
PV Status	allocatable
Total PE / Free PE	2559 / 2559

h. Fusiónar grupos de volúmenes

La fusión de dos VG se realiza con el comando `vgmerge`.

Sintaxis

```
vgmerge <nombre_de_VG_objetivo> <nombre_de_VG_a_fusionar>
```

Este comando no tiene opciones.

Ejemplo con Debian

El objetivo es poner el contenido del VG `devVG` en `appVG`:

```
# vgmerge appVG devVG
Volume group "devVG" successfully merged into "appVG"
```

Verificamos:

```
# vgdisplay -v appVG
Using volume group(s) on command line
Finding volume group "appVG"
--- Volume group ---
VG Name                appVG
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   3
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0

Open LV                 0

Max PV                 0
```

```
Cur PV          3
Act PV          3
VG Size        29,99 GiB
PE Size        4,00 MiB
Total PE       7677
Alloc PE / Size 0 / 0
Free PE / Size  7677 / 29,99 GiB
VG UUID        xdeSxK-e2Lu-YKU1-bTMi-XN1K-mJkA-p2A6Fm
```

--- Physical volumes ---

```
PV Name          /dev/sdb
PV UUID          rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

```
PV Name          /dev/sdc
PV UUID          m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

```
PV Name          /dev/sdd
PV UUID          msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

i. Renombrar un grupo de volúmenes

Podemos cambiar el nombre de un VG con el comando `vgrename`.

Sintaxis

```
vgrename <nombre_de_VG> <nuevo_nombre_de_VG>
```

Este comando no tiene opciones.

Ejemplo con Debian

El objetivo es cambiar el nombre del VG `appVG` a `progVG`:

```
# vgrename appVG progVG
```

```
Volume group "appVG" successfully renamed to "progVG"
```

Verificamos:

```
# vgsdisplay -s
```

```
"progVG" 29,99 GiB [0          used / 29,99 GiB free]
```

```
"SystemVG" 50,00 GiB [50,00 GiB used / 0    free]
```

4. Administración de volúmenes lógicos

Los comandos que gestionan los volúmenes físicos empiezan por `lv` y se almacenan en la carpeta `/sbin`:

```
# ls /sbin/lv[!m]*
```

```
/sbin/lvchange  /sbin/lvdisplay  /sbin/lvremove  /sbin/lvs
```

```
/sbin/lvconvert /sbin/lvextend  /sbin/lvrename  /sbin/lvscan
```

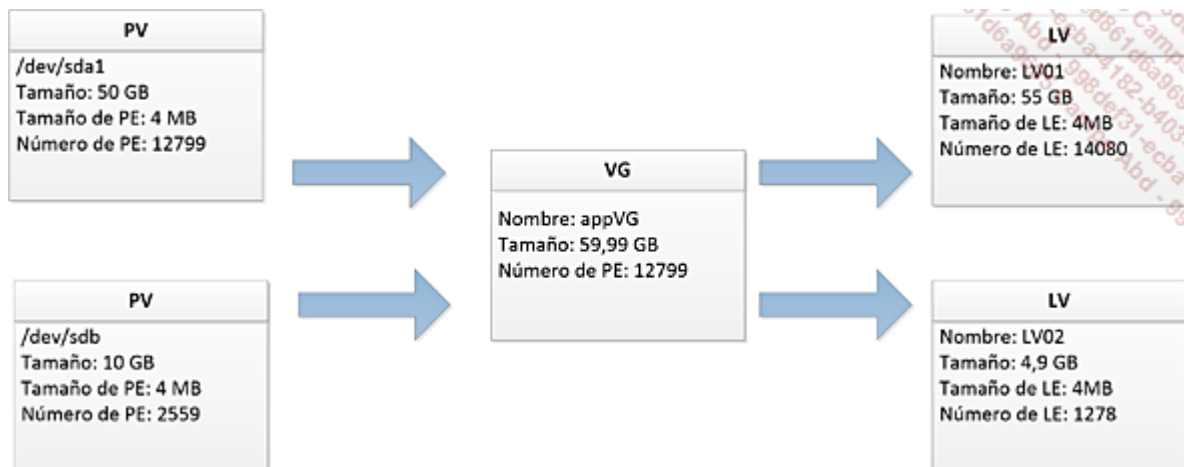
```
/sbin/lvcreate  /sbin/lvreduce  /sbin/lvresize
```

El `[!m]` excluye en el patrón la letra `m` del tercer carácter. De esta forma, evitamos obtener los comandos que comienza por `lvm` en la lista.

a. Crear volúmenes lógicos en modo lineal

Un volumen lineal es una combinación de varios PV en un LV.

Por ejemplo, si tenemos dos PV, uno de 50 GB y otro de 10 GB, podemos concebir un LV de 60 GB. El almacenamiento físico se fusiona de esta forma.



Los LE de 1 a 12799 corresponden a los PE del primer PV y aquellos que van de 12800 a 14080 corresponden a los PE del segundo PV. Al final, tenemos un único dispositivo con 14080 LE.

El comando `lvcreate` crea los LV.

Sintaxis

```
lvcreate <opciones> <nombre_de_VG>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
<code>-n <nombre_LV></code>	<code>--name <nombre_LV></code>	Define el nombre del LV.
<code>-L <tamaño></code>	<code>--size <tamaño></code>	Define el tamaño del LV.
<code>-l <valor_%></code>	<code>--extents <valor_%></code>	% utilizado por el LV en el espacio total del VG.

Ejemplo con Debian

Mostrar las características del VG `progVG`:

```
# vgdisplay -v appVG
```



```

Using volume group(s) on command line
Finding volume group "appVG"
--- Volume group ---
VG Name                appVG
System ID
Format                 lvm2
Metadata Areas        3
Metadata Sequence No  4
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                0
Open LV                0
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                29,99 GiB
PE Size                4,00 MiB
Total PE               7677
Alloc PE / Size        0 / 0
Free PE / Size         7677 / 29,99 GiB
VG UUID                xdeSxK-e2Lu-YKU1-bTMi-XN1K-mJkA-p2A6Fm

--- Physical volumes ---
PV Name                /dev/sdb
PV UUID                rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status              allocatable
Total PE / Free PE    2559 / 2559

PV Name                /dev/sdc
PV UUID                m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status              allocatable
Total PE / Free PE    2559 / 2559

```

PV Name	/dev/sdd
PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
PV Status	allocatable
Total PE / Free PE	2559 / 2559

Crear un LV de 18 GB llamado dataLV:

```
# lvcreate -n dataLV -L 18G progVG
Logical volume "dataLV" created
```

Verificamos:

```
# vgdisplay -v progVG
Using volume group(s) on command line
Finding volume group "progVG"
--- Volume group ---
VG Name                progVG
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   5
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               0
Max PV                0
Cur PV                3
Act PV                3
VG Size                29,99 GiB

PE Size                4,00 MiB

Total PE              7677
Alloc PE / Size       4608 / 18,00 GiB
```

```
Free PE / Size      3069 / 11,99 GiB
VG UUID             xdeSxK-e2Lu-YKU1-bTMi-XN1K-mJkA-p2A6Fm
```

--- Logical volume ---

```
LV Path              /dev/progVG/dataLV
LV Name              dataLV
VG Name              progVG
LV UUID              RbdfcJ-vVcM-hLZ6-qjSg-9K0B-ulKK-dC2UQa
LV Write Access      read/write
LV Creation host, time debsrv, 2013-08-17 23:16:00 +0200
LV Status             available
# open                0
LV Size               18,00 GiB
Current LE            4608
Segments              2
Allocation            inherit
Read ahead sectors    auto
- currently set to    256
Block device          254:4
```

--- Physical volumes ---

```
PV Name              /dev/sdb
PV UUID              rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status             allocatable
Total PE / Free PE    2559 / 0
```

```
PV Name              /dev/sdc
PV UUID              m68YAv-0BQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status             allocatable
Total PE / Free PE    2559 / 510
```

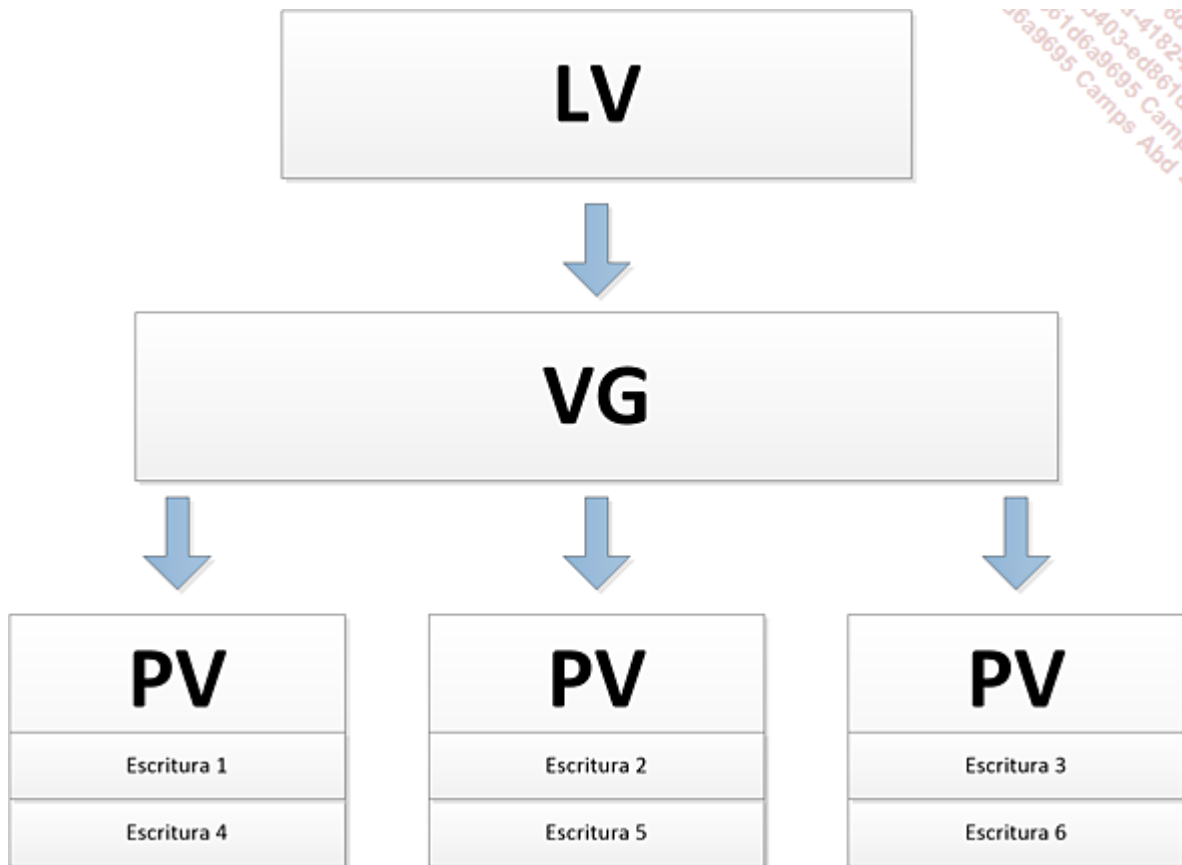
```
PV Name              /dev/sdd
```

PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-n6Uno7
PV Status	allocatable
Total PE / Free PE	2559 / 2559

DataLV tiene 4608 LE asignados entre los 2559 PE de /dev/sdb y los 2049 PE de /dev/sdc. Quedan en este último 510 PE. Todos los PE de /dev/sdd están libres.

b. Crear volúmenes lógicos en modo stripe

Creando un LV en modo stripe, los datos se escriben en los PV del VG con una técnica llamada round-robin. Esto mejora el rendimiento de lectura y escritura de los datos.



Se necesitan dos opciones adicionales para el comando lvcreate:

Opción		Descripción
corta	larga	
-i <valor>	--stripes <valor>	Define el número de bandas (stripes). El valor debe ser igual al número de PV.
-I <valor>	--stripesize <valor>	Define el número de KB para la granularidad de las bandas (stripes). No debe exceder el tamaño del PE.

Ejemplo con Debian

Antes de comenzar, eliminamos progVG:

```
# vgremove progVG
Volume group "progVG" successfully removed
```

Verificamos:

```
# vgs
VG          #PV  #LV  #SN  Attr      VSize  VFree
SystemVG    1    4    0   wz--n-   50,00g    0
```

Luego creamos el VG dataVG como sigue:

```
# vgcreate dataVG /dev/sdb /dev/sdc /dev/sdd
Volume group "dataVG" successfully created
```

El VG es de la siguiente forma:

```
# vgsdisplay -v dataVG
Using volume group(s) on command line
Finding volume group "dataVG"
--- Volume group ---
VG Name              dataVG
System ID
Format                lvm2
Metadata Areas       3
```

```

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               3
Act PV                3
VG Size               29,99 GiB
PE Size               4,00 MiB
Total PE              7677
Alloc PE / Size       0 / 0
Free PE / Size        7677 / 29,99 GiB
VG UUID               WtJTkp-2AM8-Kphw-UTwp-CsFq-FmWI-b39o7T

```

--- Physical volumes ---

```

PV Name                /dev/sdb
PV UUID                rDOPej-jvBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status              allocatable
Total PE / Free PE    2559 / 2559

```

```

PV Name                /dev/sdc
PV UUID                m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status              allocatable
Total PE / Free PE    2559 / 2559

```

```

PV Name                /dev/sdd
PV UUID                msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-6mFa3U
PV Status              allocatable
Total PE / Free PE    2559 / 2559

```

Crear el LV websiteLV de 25 GB en modo stripe en los tres PV del VG dataVG:

<https://dogramcode.com/libros-sistemas>

```
# lvcreate -L 25G -i3 -n website1LV dataVG
Using default stripesize 64,00 KiB
Rounding size (6400 extents) up to stripe boundary size (6402
extents)
Logical volume "website1LV" created
```

Verificamos:

```
# vgdisplay -v progVG
Using volume group(s) on command line
Finding volume group "progVG"
--- Volume group ---
VG Name                progVG
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   2
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               0
Max PV                0
Cur PV               3
Act PV               3
VG Size               29,99 GiB
PE Size              4,00 MiB
Total PE             7677
Alloc PE / Size       6402 / 25,01 GiB
Free PE / Size       1275 / 4,98 GiB
VG UUID              xdeSxK-2AM8-t8zW-bTMi-PC5j-mJkA-b39o7T

--- Logical volume ---
LV Path                /dev/progVG/dataLV
```

LV Name	dataLV
VG Name	progVG
LV UUID	RbdfcJ-vVcM-hLZ6-qjSg-9K0B-ulKK-B6xHf3
LV Write Access	read/write
LV Creation host, time	debsrv, 18/08/2013 11:24:16 +0200
LV Status	available
# open	0
LV Size	25,01 GiB
Current LE	6402
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	768
Block device	254:4

--- Physical volumes ---

PV Name	/dev/sdb
PV UUID	rDOPej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status	allocatable
Total PE / Free PE	2559 / 425

PV Name	/dev/sdc
PV UUID	m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status	allocatable
Total PE / Free PE	2559 / 425

PV Name	/dev/sdd
PV UUID	msidG9-XtZI-cKdo-FG4q-oxEj-zYqf-6mFa3U
PV Status	allocatable
Total PE / Free PE	2559 / 425

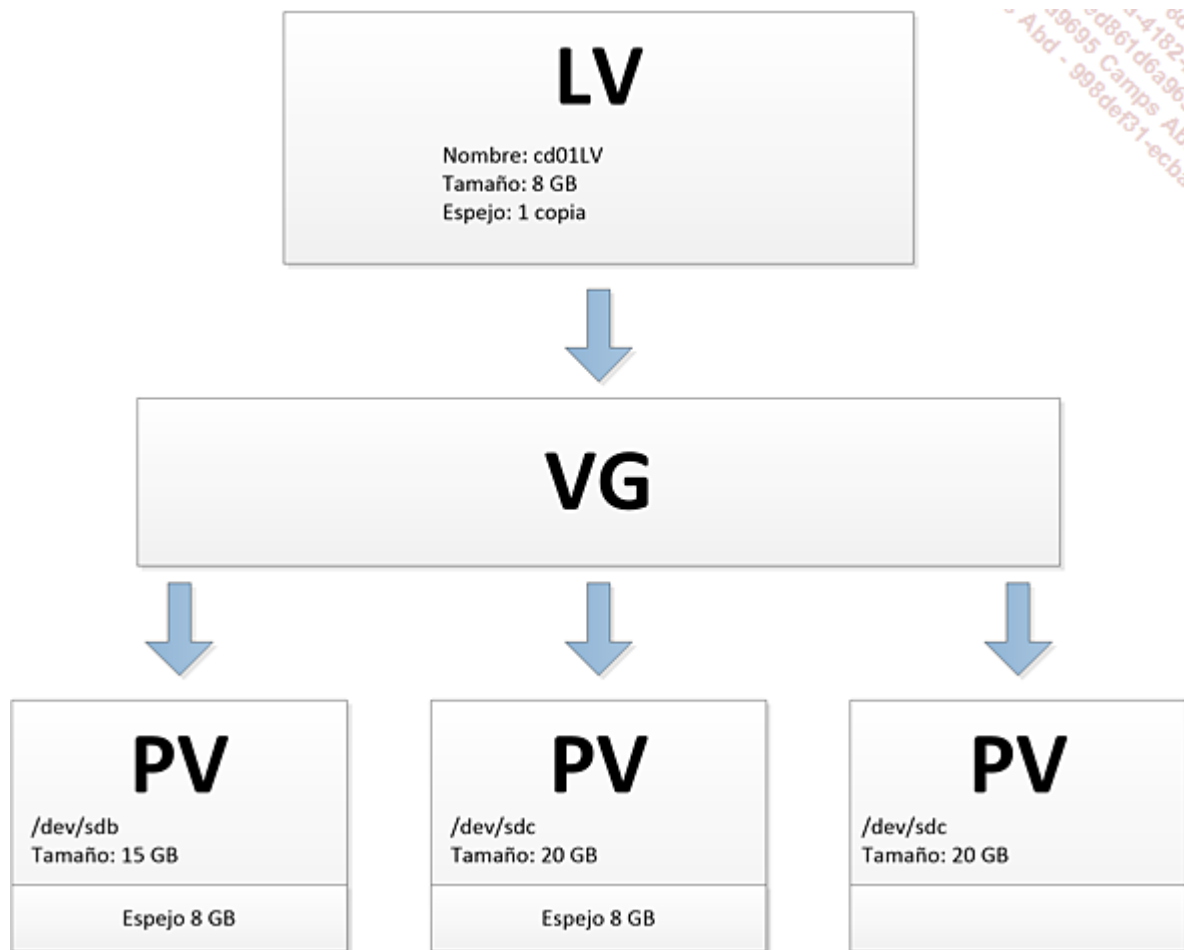
El LV websiteLV tiene 6402 PE, es decir, 2034 PE respectivamente en los tres PV: $(2559-425)*3=6402$.

c. Crear volúmenes lógicos en modo espejo

El principio de un espejo es administrar copias idénticas de datos en diferentes PV. Cuando los datos se escriben en un PV, estos se duplican en otro PV. El objetivo es protegerse ante los fallos de dispositivos.

LVM soporta varias copias en espejo.

Si un PV del espejo falla, el LV se convierte en un LV lineal sin dejar de estar accesible.



Se requiere una opción adicional para el comando lvcreate:

Opción		Descripción
corta	larga	
-m <valor>	--mirrors <valor>	Crea un volumen lógico en espejo con un número de copias definidas en <valor>.

Ejemplo con Debian

Antes de comenzar, eliminamos dataVG:

```
# vgremove dataVG
```

```
Do you really want to remove volume group "dataVG" containing 1
logical volumes? [y/n]:
```

Respondemos y.

```
Do you really want to remove active logical volume website1LV?
[y/n]: y
```

Respondemos y.

```
Logical volume "website1LV" successfully removed
Volume group "dataVG" successfully removed
```

Verificamos:

```
# vgs
```

```
VG          #PV  #LV  #SN  Attr      VSize  VFree
SystemVG    1    4    0   wz--n-   50,00g    0
```

Luego creamos el VG dataVG como sigue:

```
# vgcreate dataVG /dev/sdb /dev/sdc /dev/sdd
Volume group "dataVG" successfully created
```

El VG es de la siguiente forma:

```
# vgsdisplay -v dataVG
```

```
Using volume group(s) on command line
```

```
Finding volume group "dataVG"
```

--- Volume group ---

VG Name	dataVG
System ID	
Format	lvm2
Metadata Areas	3
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	3
Act PV	3
VG Size	29,99 GiB
PE Size	4,00 MiB
Total PE	7677
Alloc PE / Size	0 / 0
Free PE / Size	7677 / 29,99 GiB
VG UUID	Nahco1-VkfC-mNqa-mLIW-cRny-4L3o-DAQLCz

--- Physical volumes ---

PV Name	/dev/sdb
PV UUID	rDOPej-jvBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status	allocatable
Total PE / Free PE	2559 / 2559
PV Name	/dev/sdc
PV UUID	m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status	allocatable
Total PE / Free PE	2559 / 2559

PV Name	/dev/sdd
PV UUID	JXYwue-3IuZ-dI8V-oXOD-LZFY-asgh-6mFa3U
PV Status	allocatable
Total PE / Free PE	2559 / 2559

Creamos el LV crmLV de 8 GB:

```
# lvcreate -L 8G -m1 -n crmLV dataVG
Logical volume "crmLV" created
```

Verificamos:

```
# vgdisplay -v dataVG
Using volume group(s) on command line
Finding volume group "dataVG"
--- Volume group ---
VG Name                dataVG
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   3
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               0
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                29,99 GiB
PE Size                4,00 MiB
Total PE               7677
Alloc PE / Size        4097 / 16,00 GiB

Free PE / Size         3580 / 13,98 GiB
```

```

VG UUID                      EQtB02-C2M6-ToC5-VX3M-FZ3Z-mMpu-NxSrc9

--- Logical volume ---
LV Path                      /dev/dataVG/crmLV
LV Name                      crmLV
VG Name                      dataVG
LV UUID                      65udHU-NlWr-gN2M-igmg-Rg7Z-xYtT-GucBJd
LV Write Access              read/write
LV Creation host, time       debsrv, 2013-08-18 11:43:19 +0200
LV Status                    available
# open                      0
LV Size                      8,00 GiB
Current LE                   2048
Mirrored volumes            2
Segments                    1
Allocation                  inherit
Read ahead sectors          auto
- currently set to          256
Block device                 254:7

--- Physical volumes ---
PV Name                      /dev/sdb
PV UUID                      rDOPej-jvBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status                    allocatable
Total PE / Free PE          2559 / 511

PV Name                      /dev/sdc
PV UUID                      m68YAv-0BQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status                    allocatable
Total PE / Free PE          2559 / 511

PV Name                      /dev/sdd
PV UUID                      JXYwue-3IuZ-dI8V-oXOD-LZFY-asgh-6mFa3U

```

```
PV Status          allocatable
Total PE / Free PE 2559 / 2558
```

Tenemos dos volúmenes en espejo, uno en /dev/sdb y otro en /dev/sdc.

Para reparar un espejo después de un fallo del disco, basta con teclear:

```
# lvconvert -m1 dataVG/crmLV
```

d. Crear instantáneas de volumen lógico

LVM crea imágenes virtuales de volúmenes en un momento dado sin interrumpir el servicio. Si se efectúa un cambio en la fuente, LVM crea entonces una copia del bloque modificado en el snapshot (instantánea). La instantánea no copia el LV fuente de forma completa. Almacenará solo los cambios. Es la razón por la que se define el tamaño. Para observar su evolución, utilizamos `lvdisplay`, que explicamos más adelante.

El interés de hacer un snapshot de un volumen es que nos permite montarlo más tarde para probar un software. Si el resultado no resulta satisfactorio, podemos desmontar para borrar y luego volver a montar el sistema de archivos de origen. O por el contrario, podemos desear que las modificaciones introducidas en el snapshot figuren en el LV de origen.



Las instantáneas no están soportados en los LV en espejo.

Se necesita una opción adicional para crear un snapshot con `lvcreate`:

Opción		Descripción
corta	larga	
-s	--snapshot	Crear un snapshot.

Ejemplo con Debian

Antes de comenzar, eliminamos dataVG:

```
# vgremove dataVG
```

```
Do you really want to remove volume group "dataVG" containing 1
logical volumes? [y/n]:
```

Respondemos y.

```
Do you really want to remove active logical volume website1LV?
```

```
[y/n]: y
```

```
Logical volume "website1LV" successfully removed
```

```
Volume group "dataVG" successfully removed
```

Verificamos:

```
# vgs
```

VG	#PV	#LV	#SN	Attr	VSize	VFree
SystemVG	1	4	0	wz--n-	50,00g	0

Luego creamos un nuevo VG llamado appVG como sigue:

```
# vgcreate dataVG /dev/sdb /dev/sdc /dev/sdd
```

```
Volume group "dataVG" successfully created
```

El VG es de la siguiente forma:

```
# vgsdisplay -v appVG
```

```
Using volume group(s) on command line
```

```
Finding volume group "appVG"
```

```
--- Volume group ---
```

```
VG Name                appVG
```

```
System ID
```

```
Format                 lvm2
```

```
Metadata Areas         3
```

```
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          3
Act PV           3
VG Size          29,99 GiB
PE Size          4,00 MiB
Total PE         7677
Alloc PE / Size  0 / 0
Free PE / Size   7677 / 29,99 GiB
VG UUID          qD4NBe-RzMM-LN32-wc0K-P49c-4Lgf-w3ZnTW
```

--- Physical volumes ---

```
PV Name          /dev/sdb
PV UUID          rD0Pej-jVBY-OQhj-93qu-Rx9G-qXXA-eh2mNX
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

```
PV Name          /dev/sdc
PV UUID          m68YAv-OBQx-0edw-TNS9-x3az-Mj6D-D70wjb
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

```
PV Name          /dev/sdd
PV UUID          JXYwue-3IuZ-dI8V-oXOD-LZFY-asgh-6mFa3U
PV Status        allocatable
Total PE / Free PE 2559 / 2559
```

Crear un LV progLV de 15 GB en el VG appVG:

```
# lvcreate -L 15G -n progLV appVG
Logical volume "progLV" created
```


Ahora, vamos a crear un snapshot de progLV:

```
# lvcreate -L 10g -s -n progLV_18082013 /dev/appVG/progLV
Logical volume "progLV_18082013" created
```

Si el tamaño del snapshot es demasiado pequeño, podemos cambiar su tamaño. Por ejemplo, para añadir 1 GB:

```
# lvresize -L +1G /dev/appVG/progLV_18082013
Extending logical volume progLV_18082013 to 11,00 GiB
Logical volume progLV_18082013 successfully resized
```

También podemos reducir su tamaño:

```
# lvresize -L -2G /dev/appVG/progLV_18082013
WARNING: Reducing active logical volume to 9,00 GiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce progLV_18082013? [y/n]:
```

Respondemos y:

```
Reducing logical volume progLV_18082013 to 9,00 GiB
Logical volume progLV_18082013 successfully resized
```

Las modificaciones efectuadas en el snapshot pueden integrarse definitivamente en el LV fuente con `lvconvert --merge`:

```
# lvconvert --merge /dev/appVG/progLV_18082013
Merging of volume progLV_18082013 started.
progLV: Merged: 0,0%
Merge of snapshot into logical volume progLV has finished.
Logical volume "progLV_18082013" successfully removed
```

e. Mostrar los volúmenes lógicos

Disponemos de tres comandos para mostrar los LV existentes:

- `lvs` devuelve la información de los LV.

- `lvscan` enumera los LG y actualiza la caché.
- `lvdisplay` muestra las propiedades de un LV.

Sintaxis

```
lvs
```

No se requieren opciones para `lvs`.

Sintaxis

```
lvscan <opciones>
```

No se requieren opciones para `lvscan`.

Sintaxis

```
lvdisplay <opciones> <nombre_de_VG>
```

Opción útil:

Opción		Descripción
corta	larga	
-m	--maps	Mostrar los LE, PV y PE.

Si se omite el nombre del VG, `lvdisplay` mostrará todos los VG con sus LV respectivos.

Ejemplo con Debian

Mostrar los LV con `lvs`:

```
# lvs
LV      VG      Attr      Lsize   Pool    Origin   Data%   Move...
homeLV   SystemVG -wi-ao--  35,39g
optLV    SystemVG -wi-ao--  168,00m
rootLV   systemVG -wi-ao--  13,97g
swapLV   SystemVG -wi-ao--  488,00m
progLV   appVG    -wi-a---  15,00g
```

Con `lvscan`:

```
# lvscan
ACTIVE                '/dev/appVG/progLV' [15,00 GiB] inherit
```

ACTIVE	'/dev/SystemVG/swapLV' [488,00 MiB] inherit
ACTIVE	'/dev/SystemVG/rootLV' [13,97 GiB] inherit
ACTIVE	'/dev/SystemVG/homeLV' [35,39 GiB] inherit
ACTIVE	'/dev/SystemVG/optLV' [168,00 MiB] inherit

Con lvsdisplay:

```
# lvsdisplay -m appVG
--- Logical volume ---
LV Path                /dev/appVG/progLV
LV Name                 progLV
VG Name                 appVG
LV UUID                 OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access         read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status                available
# open                  0
LV Size                 15,00 GiB
Current LE              3840
Segments                2
Allocation              inherit
Read ahead sectors      auto
- currently set to      256
Block device            254:2

--- Segments ---
Logical extent 0 to 2558:
  Type                  linear
  Physical volume        /dev/sdb
  Physical extents       0 to 2558

Logical extent 2559 to 3839:
  Type                  linear
  Physical volume        /dev/sdc
```

Physical extents 0 to 1280

f. Aumentar el tamaño de un volumen lógico

El comando `lvextend` permite aumentar el tamaño de un LV. Tenemos que contar, por supuesto, con espacio disponible o añadir un PV en el VG.

Sintaxis

```
lvextend <opciones> <nombre_del_LV>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
<code>-L</code> <valor>	<code>--size</code> <valor>	Aumenta el tamaño del LV en el <valor> especificado. La unidad por defecto es el MB. Podemos usar otra unidad: G (GB), T (TB), P (PB), E (EB). Tenemos dos posibilidades: La primera es indicar el valor final del LV: <code>lvextend -L10G nombre_de_LV</code> La segunda es definir el valor de aumento: <code>lvextend -L+2G nombre_de_LV</code>
<code>-r</code>	<code>--resizefs</code>	Redimensionar el tamaño de los sistemas de archivo ext2, ext3 y ext4

Ejemplo con Debian

El objetivo es aumentar en 2 GB el LV `progLV`.

Mostrar el tamaño actual de `progLV` del VG `appVG`:

```
# lvsdisplay appVG
--- Logical volume ---
LV Path                /dev/appVG/progLV
LV Name                 progLV
VG Name                 appVG
LV UUID                 OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access         read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status                available
# open                  0
LV Size                 15,00 GiB
Current LE              3840
Segments                2
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            254:2
```

Hacen falta 15 GB.

Procedemos a aumentar en 2 GB el LV:

```
# lvextend -L+2G /dev/appVG/progLV -r
Extending logical volume progLV to 17,00 GiB
Logical volume progLV successfully resized
```

O utilizando la otra posibilidad:

```
# lvextend -L17G /dev/appVG/progLV -r
Extending logical volume progLV to 17,00 GiB
Logical volume progLV successfully resized
```

Verificamos el resultado:

```
# lvsdisplay appVG
--- Logical volume ---
```

```

LV Path                /dev/appVG/progLV
LV Name                progLV
VG Name                appVG
LV UUID                OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access        read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status              available
# open                 0
LV Size                17,00 GiB
Current LE             4352
Segments               2
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device           254:2

```

g. Reducir el tamaño de un volumen lógico

Si podemos aumentar el tamaño de un LV, también lo podemos reducir. El comando `lvreduce` permite realizar esta operación.

Sintaxis

```
lvreduce <opciones> <nombre_de_LV>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
<code>-L <valor></code>	<code>--size <valor></code>	Reduce el tamaño del LV en el <valor> especificado. La unidad por defecto es el MB. Podemos usar otra unidad: G (GB), T (TB), P (PB), E (EB).
<code>-l <valor></code>	<code>--extents <valor></code>	Reduce el tamaño del LV el número de LE especificado en <valor>. El signo «-» delante de <valor> permite restar.

-r	--resizefs	Redimensionar el tamaño de los sistemas de archivo ext2, ext3 y ext4.

Ejemplo con Debian

El objetivo es reducir en 2 GB el LV progLV.

Mostrar el tamaño actual de progLV del VG appVG:

```
# lvsdisplay appVG
--- Logical volume ---
LV Path                /dev/appVG/progLV
LV Name                 progLV
VG Name                appVG
LV UUID                OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access        read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status               available
# open                 0
LV Size                 17,00 GiB
Current LE              4352
Segments                2
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device           254:2
```

Hacen falta 17 GB.

Reducir 3 GB especificando el nuevo tamaño del LV:

```
# lvreduce -L14G /dev/appVG/progLV -r
New size given (3584 extents) not larger than existing size
(4352 extents)
Run `lvextend --help' for more information.

root@debsrv:/home/bob# lvreduce -L14G /dev/appVG/progLV
WARNING: Reducing active logical volume to 14,00 GiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
```

```
Do you really want to reduce progLV? [y/n]:
```

O bien reduciendo el número de LE cuyo tamaño es de 4 MB por defecto. Hay que retirar 767 LE ($767 \times 4 = 3068$) para alcanzar 14 GB:

```
# lvreduce -l -767 /dev/appVG/progLV -r
WARNING: Reducing active logical volume to 14,00 GiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce progLV? [y/n]:
```

Respondemos y para confirmar la reducción:

```
Reducing logical volume progLV to 14,00 GiB
Logical volume progLV successfully resized
```

Verificamos:

```
# lvsdisplay appVG
--- Logical volume ---
LV Path                /dev/appVG/progLV
LV Name                 progLV
VG Name                 appVG
LV UUID                 OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access         read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status                available
# open                  0
LV Size                 14,00 GiB
Current LE               3584
```

h. Renombrar un volumen lógico

El comando `lvrename` permite renombrar un LV.

Sintaxis

```
lvrename <ruta/nombre_de_LV> <ruta/nuevo_nombre_de_LV>
```

O bien:

```
lvrename <nombre_de_VG> <nombre_de_LV> <nuevo_nombre_de_LV>
```

Este comando no tiene opciones.

Ejemplo con Debian

El objetivo es cambiar el nombre de progLV a softLV:

```
# lvrename /dev/appVG/progLV /dev/appVG/softLV
Renamed "progLV" to "softLV" in volume group "appVG"
```

O bien:

```
# lvrename appVG progLV softLV
Renamed "progLV" to "softLV" in volume group "appVG"
```

Verificamos:

```
# lvsdisplay appVG
--- Logical volume ---
LV Path                /dev/appVG/softLV
LV Name                 softLV
VG Name                 appVG
LV UUID                 OH1Di1-6DJH-sq0n-yAEc-lRoK-3G0A-EIgRVK
LV Write Access         read/write
LV Creation host, time debsrv, 2013-08-18 14:35:04 +0200
LV Status                available
# open                  0
LV Size                 14,00 GiB
Current LE               3585
Segments                2

Allocation              inherit
Read ahead sectors      auto
- currently set to      256
```

Block device

254:2

i. Eliminar un volumen lógico

El comando `lvremove` elimina un LV inactivo. Si el LV se encuentra en uso, por lo tanto montado, es necesario desmontarlo.

Sintaxis

```
lvremove <ruta/nombre_de_LV>
```

Este comando no tiene opciones.

Ejemplo con Debian

Ya que el LV `softLV` no está montado en el sistema de archivos, lo podremos eliminar:

```
# lvremove /dev/appVG/softLV
```

```
Do you really want to remove active logical volume softLV? [y/n]:
```

Respondemos y para confirmar la reducción:

```
Logical volume "softLV" successfully removed
```

Verificamos:

```
# lvdisplay appVG
```

El VG `appVG` está ahora vacío.

5. Mover los datos

El uso del comando `pvmove` permite desplazar los datos mientras el sistema está en funcionamiento.

Sintaxis

```
pvmove <opciones> <fuente> <destino>
```

No hay opciones.

Ejemplo con Debian

El objetivo es desplazar los datos del LV `docLV` del VG `officeVG` para sustituir el PV `/dev/sdb` por `/dev/sdd`.

Eliminamos el VG `appVG`:

```
# vgremove appVG
```

```
Volume group "appVG" successfully removed
```

Creamos un nuevo VG office VG con los dos PV /dev/sdb y /dev/sdc:

```
# vgcreate officeVG /dev/sdb /dev/sdc  
Volume group "officeVG" successfully created
```

Luego creamos el LV lineal docLV de 8 GB:

```
# lvcreate officeVG -L8G -n docLV  
Logical volume "docLV" created
```

Verificar que hemos creado officeVG con dos PV:

```
# vgdisplay -v officeVG  
Using volume group(s) on command line  
Finding volume group "officeVG"  
--- Volume group ---  
VG Name                officeVG  
System ID  
Format                 lvm2  
Metadata Areas         2  
Metadata Sequence No   2  
VG Access              read/write  
VG Status              resizable  
MAX LV                 0  
Cur LV                1  
Open LV                0  
Max PV                 0  
Cur PV                2  
Act PV                 2  
VG Size                19,99 GiB  
PE Size                4,00 MiB  
  
Total PE               5118  
  
Alloc PE / Size        2048 / 8,00 GiB
```

```

Free PE / Size      3070 / 11,99 GiB
VG UUID             Pvh2em-FmhX-MtrP-cLkB-9ftQ-gbAU-IUj1NZ

--- Logical volume ---
LV Path              /dev/officeVG/docLV
LV Name              docLV
VG Name              officeVG
LV UUID              1Hjc5g-hE7K-wdFV-Lqfi-5Pz0-LB0h-f2Ghh8
LV Write Access      read/write
LV Creation host, time debsrv, 2013-08-20 16:34:40 +0200
LV Status             available
# open                0
LV Size              8,00 GiB
Current LE            2048
Segments              1
Allocation            inherit
Read ahead sectors    auto
- currently set to    256
Block device          254:2

--- Physical volumes ---
PV Name              /dev/sdb
PV UUID              RVk0aN-xOq0-tJiz-Ld0y-SYm0-hriu-3QLEEC
PV Status             allocatable
Total PE / Free PE   2559 / 511

PV Name              /dev/sdc
PV UUID              gXJv6W-2UrM-v190-9hKb-hDFF-qLyQ-G3TRbU
PV Status             allocatable
Total PE / Free PE   2559 / 2559

```

El LV docLV posee 2048 LE en /dev/sdb (2559-511=2048).

Formatear docLV en ext4:

```
# mkfs -t ext4 /dev/officeVG/docLV
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb1 contiene un sistema de ficheros ext3
fecha de creación Sun May 8 17:25:17 2016
Se está creando El sistema de ficheros con 1310720 4k bloques y
327680 nodos-i
UUID del sistema de ficheros: bb195edc-2a59-4419-9fa1-71feade53f95
Respaldo del superbloque guardado en los bloques:
32768, 98304, 163840, 229376, 294912, 819200, 884736
Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (32768 bloques): hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: 0/4hecho
```

Crear una carpeta \$HOME/DOC:

```
# mkdir $HOME/doc
```

Montar el volumen docLV en la carpeta \$HOME/doc:

```
# mount /dev/officeVG/docLV $HOME/doc
```

Crear un archivo \$HOME/data/lista.txt:

```
# ls /* > $HOME/doc/lista.txt
```

Vamos a mover los datos del LV almacenados en el PV /dev/sdb al PV /dev/sdc:

```
# pvmove /dev/sdb /dev/sdc
/dev/sdb: Moved: 0,4%
/dev/sdb: Moved: 81,6%
/dev/sdb: Moved: 100,0%
```

Los 2048 PE de docLV ya no se encuentran en /dev/sdb sino en /dev/sdc:

```
# vgdisplay -v officeVG
Using volume group(s) on command line
Finding volume group "officeVG"
```

--- Volume group ---

VG Name	officeVG
System ID	
Format	lvm2
Metadata Areas	2
Metadata Sequence No	5
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	19,99 GiB
PE Size	4,00 MiB
Total PE	5118
Alloc PE / Size	2048 / 8,00 GiB
Free PE / Size	3070 / 11,99 GiB
VG UUID	Pvh2em-FmhX-MtrP-cLkB-9ftQ-gbAU-IUj1NZ

--- Logical volume ---

LV Path	/dev/officeVG/docLV
LV Name	docLV
VG Name	officeVG
LV UUID	1Hjc5g-hE7K-wdFV-Lqfi-5Pz0-LB0h-f2Ghh8
LV Write Access	read/write
LV Creation host, time	debsrv, 2013-08-20 16:34:40 +0200
LV Status	available
# open	1
LV Size	8,00 GiB
Current LE	2048

```

Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device             254:2

--- Physical volumes ---
PV Name                  /dev/sdb
PV UUID                  RVk0aN-x0q0-tJiz-Ld0y-SYm0-hriu-3QLEEC
PV Status                allocatable
Total PE / Free PE      2559 / 2559

PV Name                  /dev/sdc
PV UUID                  gXJv6W-2UrM-v190-9hKb-hDFF-qLyQ-G3TRbU
PV Status                allocatable
Total PE / Free PE      2559 / 5

```

Desmonte /doc:

```
# umount $HOME/doc
```

Eliminamos el PV /dev/sdb del VG:

```
# vgreduce officeVG /dev/sdb
Removed "/dev/sdb" from volume group "officeVG"
```

Agregamos el PV /dev/sdd en el VG:

```
# vgextend officeVG /dev/sdd
Volume group "officeVG" successfully extended
```

Montar el volumen docLV en la carpeta \$HOME/doc:

```
# mount /dev/officeVG/docLV $HOME/doc
```

Mostrar la carpeta \$HOME/doc:

```
# ls $HOME/doc  
lista.txt  lost+found
```

El archivo lista.txt sigue aquí.

Sistemas de archivo

Un sistema de archivos (SA) o file system (FS) se encarga de gestionar la organización de los datos almacenados en dispositivos de bloque como discos duros, llaves USB, etc.

El sistema ext (Extended File System) es el primer sistema de archivos para GNU/Linux creado en 1992 por Rémy Card para paliar las muchas limitaciones del sistema de archivos utilizado hasta entonces, Minix FS de Andrew Tanenbaum. La longitud máxima del nombre de archivo era de 14 caracteres. Una partición no excedía los 64 MB...

Luego ext se sustituye por ext2, el cual se incluye en el kernel de Linux 0.99.15 en diciembre de 1993.

1. ext2

El Second Extended File System, más conocido como ext2, utiliza el bloque como unidad de almacenamiento.

Las características de ext2:

Característica	Capacidad	a. Bloques Una partición o disco formateado con ext2 está dividido en pequeños grupos de sectores llamados «bloques». Estos bloques se agrupan a continuación en unidades mayores denominadas «grupos de bloques». El tamaño de los bloques se determinará durante el formateo del disco. Esto tiene un impacto sobre el rendimiento, el tamaño máximo de un archivo y el
Tamaño máximo de archivo	16 GB - 2 TB	
Número máximo de archivos	Variable	
Tamaño máximo del nombre de archivo	255 caracteres	
Tamaño máximo de volumen	2 GB - 32 TB	
Tamaño máximo de una partición	32 TB	

tamaño máximo del sistema de archivos. Los tamaños de bloque usados habitualmente son 1 KB, 2, KB, 4 KB y 8 KB.

El comando `badblocks` permite buscar bloques defectuosos en un dispositivo de bloque, en concreto en una partición de disco.

Sintaxis

```
badblocks <opciones> <partición>
```

Opciones necesarias:

Opción corta	Descripción
-n	Utiliza el modo no-destructivo de lectura/escritura (read/write).
-v	Modo largo (verbose).
-s	Muestra el progreso en %.

Ejemplo con Debian

Analizar la partición en forma de lectura-escritura no destructiva `/dev/sdb1`:

```
# badblocks -nvs /dev/sdb1
Revisando los bloques dañados en modo lectura-escritura no
destructivo
Del bloque 0 al 5242879
Revisando los bloques dañados (prueba de lectura-escritura no
destructiva)
Probando con un patrón aleatorio:  0.00% hecho, 0:00 transcurrido.
(0/0/0 error hecho
Paso terminado, se encontraron 0 bloques dañados. (0/0/0 errores)
```

En este ejemplo, ninguno de los bloques es defectuoso.

b. Superbloque

El superbloque (superblock) contiene todos los datos relativos a la configuración del sistema de archivos como el número total de inodos, los bloques del sistema de archivos y el número de inodos disponibles, etc. Existen varias copias del superbloque en el disco.

El comando `dumpe2fs` muestra información procedente del superbloque.

Sintaxis

```
dumpe2fs <opción> <partición>
```

Opción necesaria:

Opción corta	Descripción
-h	Muestra solo la información del superbloque.

Ejemplo con Debian

Mostrar información del superbloque de la partición /dev/sdb1:

```
# dumpe2fs -h /dev/sdb1
dumpe2fs 1.42.5 (29-Jul-2012)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         d83eb92e-9575-4756-b606-c653c1c4702d
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      ext_attr resize_inode dir_index filetype
                           sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              183264
Block count:              732160
Reserved block count:     36608
Free blocks:              719574
Free inodes:              183253
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      178
```

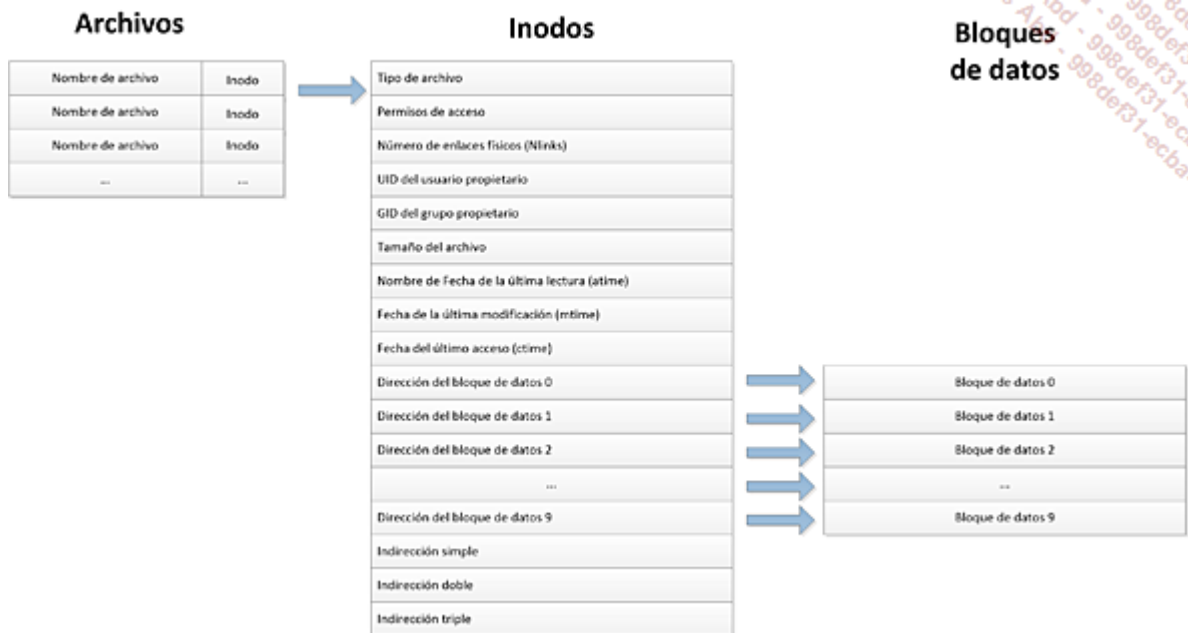
Blocks per group:	32768
Fragments per group:	32768
Inodes per group:	7968
Inode blocks per group:	498
Filesystem created:	Sun Sep 1 16:33:34 2013
Last mount time:	n/a
Last write time:	Sun Sep 1 16:33:34 2013
Mount count:	0
Maximum mount count:	-1
Last checked:	Sun Sep 1 16:33:34 2013
Check interval:	0 (<none>)
Reserved blocks uid:	0 (user root)
Reserved blocks gid:	0 (group root)
First inode:	11
Inode size:	256
Required extra isize:	28
Desired extra isize:	28
Default directory hash:	half_md4
Directory Hash Seed:	b1ace00e-8575-4a7a-bbd8-6e6a0d0430ca

c. Inodos

El inodo (contracción de «índice» y «nodo») es una estructura de datos que contiene información sobre los archivos almacenados, como su creador, propietario, tipo de acceso, etc. Un número de inodo corresponde a un archivo en el sistema de archivos. El número máximo de inodos, y por lo tanto el número máximo de archivos y carpetas, se define durante la instalación del sistema de archivos.

Los primeros 10 campos de un total de 13 contienen direcciones de los 10 primeros bloques de datos del archivo a razón de una dirección por bloque. Si los bloques de los 10 primeros campos no son suficientes, entonces se utilizan los campos 11, 12 y 13. Funciona según un sistema de direccionado indirecto. Existen tres niveles de direccionado indirecto:

- El direccionado indirecto simple, utilizado por el campo 11.
- El direccionado indirecto doble, utilizado por el campo 12.
- El direccionado indirecto triple, utilizado por el campo 13.



El campo 11 es un simple direccionado indirecto. Apunta a un bloque de direcciones. Este bloque de direcciones contiene 256 punteros hacia bloques de datos.

Si se requiere, el sistema puede utilizar el bloque 12, que es un doble direccionado indirecto. Este bloque apunta como el campo 11 hacia un bloque de direcciones. Sin embargo, este último no apunta a 256 bloques de datos. Apunta hacia otros 256 bloques de direcciones, los cuales apuntan hacia 256 bloques de datos.

En cuanto al campo 13^a, corresponde a un direccionado indirecto triple. Apunta hacia un bloque de 256 direcciones. Cada bloque apunta a un bloque de 256 direcciones que apuntan a su vez a 256 bloques de direcciones. Estos últimos apuntan a su vez hacia 256 bloques de datos.

El tamaño máximo de un archivo puede calcularse multiplicando por 1024 bytes el número de bloques de datos total: $1024 * (10 + 256^1 + 256^2 + 256^3)$.

El comando stat es el nombre de una llamada de sistema y también de un comando que forma parte del paquete coreutils o GNU core utilities. Esta herramienta permite obtener información sobre los archivos o los directorios.

Sintaxis

```
stat <archivo>
```

Este comando no tiene opciones.

Ejemplos con Debian

```
# stat $(which ls)
  Fichero: «/bin/ls»
  Tamaño: 118280          Bloques: 232  Bloque E/S: 4096
fichero regular
Dispositivo: fe00h/65024d  Nodo-i: 227  Enlaces: 1
Acceso: (0755/-rwxr-xr-x)  Uid: (  0/  root)  Gid: (  0/  root)
Acceso: 2016-05-07 19:52:20.603205972 +0200
Modificación: 2015-03-14 16:47:04.000000000 +0100
Cambio: 2016-03-30 22:21:32.650041000 +0200
Creación: -
```

d. Crear un sistema de archivos ext2

El comando mke2fs permite formatear una partición con formato ext2 siempre que no esté montada.

Sintaxis

```
mke2fs <partición>
```

Este comando no tiene opciones.

Ejemplo con Debian

Crear un sistema de archivos ext2 en /dev/sdb1:

```
# mke2fs /dev/sdb1
root@debiansrv:/home/bob# mke2fs /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdd1 contiene un sistema de ficheros ext2
fecha de creación Mon May  9 19:59:57 2016
```

```
¿Continuar de todas formas? (s,n) s
Se está creando El sistema de ficheros con 787456 4k bloques y
197200 nodos-i
UUID del sistema de ficheros: 2211d504-ffdc-464f-9e96-460e61b8d87e
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: 0/2hecho
```

2. ext3

Desarrollado por Stephen Tweedie, ext3 es una evolución de ext2 que permite crear registros de transacciones del sistema de archivos (journaling file system). Se encuentra en la rama 2.4.15 del kernel de Linux desde noviembre de 2001.

Un sistema que puede leer y escribir en una partición ext2 puede a su vez leer y escribir en una partición ext3 sin registros de transacciones.

Las características de ext3 son idénticas a las de ext2.

a. El archivo de transacciones

Un archivo de transacciones mantiene una traza de las escrituras hasta que están terminadas, esto se hace con el fin de garantizar la integridad de los datos en caso de un fallo catastrófico.

El objetivo es poder recuperar los datos de forma sencilla y rápida en caso de una interrupción brusca del sistema operativo, ya que las particiones no se habrán sincronizado de forma correcta.

Sin el archivo de transacciones, una herramienta de recuperación de datos utilizada después de una interrupción brusca deberá recorrer todo el sistema de archivos para verificar su coherencia. Si el tamaño del sistema de archivos es significativo, la operación puede durar un tiempo importante con la posibilidad de una pérdida de datos.

b. Convertir una partición ext2 a ext3

La conversión se realiza con el comando `tune2fs` siempre que la partición se encuentre desmontada.

Sintaxis

```
tune2fs <opción> <partición>
```

Opción necesaria:

Opción corta	Descripción
-j	Añade un registro.

Ejemplo con Debian

Convertir `/dev/sdb1`:

```
# tune2fs -j /dev/sdb1
tune2fs 1.42.12 (29-Aug-2014)
Creando el nodo-i del fichero de transacciones: hecho
```

c. Crear un sistema de archivos ext3

Disponemos de los comandos `mke2fs` y `mkfs.ext3`.

Sintaxis

```
mke2fs <opción> <partición>
```

Opción necesaria:

Opción corta	Descripción	Sintaxis
-j	Añade un registro al sistema de archivos.	<code>mkfs.ext3 <partición></code>
		No hay opciones.

Ejemplo con Debian

Crear un sistema de archivos ext3 en /dev/sdb1 usando mke2fs:

```
# mke2fs -j /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb1 contiene un sistema de ficheros ext2
fecha de creación Mon May  9 20:01:28 2016
¿Continuar de todas formas? (s,n) s
Se está creando El sistema de ficheros con 787456 4k bloques y
197200 nodos-i
UUID del sistema de ficheros: a6f6e1b4-6734-4312-918a-9b36529c638a
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema
de ficheros: 0/2 hecho
```

Crear un sistema de archivos ext3 en /dev/sdb1 usando mkfs.ext3:

```
# mkfs.ext3 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb1 contiene un sistema de ficheros ext3
fecha de creación Sun May  8 19:27:10 2016
¿Continuar de todas formas? (s,n) s
Se está creando El sistema de ficheros con 786432 4k bloques y
196608 nodos-i
UUID del sistema de ficheros: 08b8acd6-5455-4f66-bd82-0bed2c4100ce
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
```



```
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema
de ficheros: hecho
```

3. ext4

ext4 mantiene la compatibilidad con su predecesor ext3. Se considera como una etapa intermedia a la espera de una nueva generación de sistemas de archivos. Está incluido en el kernel Linux 2.6.19 (29 de noviembre de 2006). A partir la versión 2.6.28 (24 de diciembre de 2008) del kernel, ext4 se considera estable.

Las características de ext4:

Características	Capacidades	La
Tamaño máximo de archivo	16 TB	
Número máximo de archivos	4 millones	
Tamaño máximo del nombre de archivo	256 bytes	
Tamaño máximo de volumen	1 EB (limitado a 16 TB para e2fsprogs)	

característica principal de ext4 es la asignación por «extensión» que permite la pre-asignación de una zona contigua a un archivo con el objetivo de minimizar la fragmentación. Se encuentra activa por defecto a partir del kernel Linux 2.6.23. Antes, se debía indicar de forma explícita durante el montaje de la partición de esta forma:

```
# mount /dev/sda1 /mnt/point -t ext4dev -o extents
```

a. Convertir una partición ext3 a ext4

La conversión se realiza con el comando `tune2fs` siempre que la partición se encuentre desmontada.

Sintaxis

```
tune2fs <opción> <partición>
```

Opción necesaria:

Opción corta	Descripción
-O	Elimina las características indicadas del sistema de archivos. uninit_bg permite al kernel inicializar la tabla de bloques y de inodos. dir_index permite utilizar los árboles binarios hash para acelerar la búsqueda en directorios de gran tamaño.

Ejemplo con Debian

Convertir a ext4 la partición /dev/sdb1:

```
# tune2fs -O extents,uninit_bg,dir_index /dev/sdb1  
tune2fs 1.42.5 (29-Jul-2012)
```

b. Crear un sistema de archivos ext4

Disponemos de los comandos mke2fs y mkfs.ext4.

Sintaxis

```
mke2fs <opción> <partición>
```

Opción necesaria:

Opción corta	Descripción
-t	Indica el tipo de sistema de archivos. Por ejemplo -t ext4.

Sintaxis

```
mkfs.ext4 <partición>
```

Este comando no tiene opciones.

Ejemplo con Debian

Crear un sistema de archivos ext4 en /dev/sdb1 usando mke2fs:

```
# mke2fs -j /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb1 contiene un sistema de ficheros ext3
fecha de creación Sun May  8 19:28:20 2016
¿Continuar de todas formas? (s,n) s
Se está creando El sistema de ficheros con 786432 4k bloques y
196608 nodos-i
UUID del sistema de ficheros: c2e0fbe7-a687-44b0-b423-28e58ef9b90d
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912
Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de
ficheros: hecho
```

Crear un sistema de archivos ext4 en /dev/sdb1 usando mkfs.ext4:

```
# mkfs.ext4 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdb1 contiene un sistema de ficheros ext4
fecha de creación Sun May  8 19:31:09 2016
¿Continuar de todas formas? (s,n) s
Se está creando El sistema de ficheros con 786432 4k bloques y
196608 nodos-i
UUID del sistema de ficheros: 8d6fadda-3904-4923-9ec1-550b22dc62e1
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de
```

Ejercicios

1. Particionado durante la instalación

1. Cree una nueva máquina virtual:

Componentes	Valores
General	Linux Debian 8 (64 bits) 768 MB de RAM
Sistema	Orden de arranque: disco duro medio óptico
Almacenamiento	Controlador CD/DVD Imagen ISO DVD1 de Debian 8 Cinco discos duros (asignados dinámicamente): Disco 1: 8 GB Disco 2: 10 GB Disco 3: 5 GB Disco 4: 10 GB Disco 5: 9 GB
Red	Adaptador 1 conectado a NAT

2. Instale Debian 8 con la configuración siguiente:

Componentes	Valores
Cuenta root	Desactivada

Cuenta bob	Sudoer Contraseña: formación
Partición Primaria	/boot : 1 GB (ext2)
Grupo de volúmenes	Nombre: sysVG Lista de PV: /dev/sda /dev/sdb
Volúmenes lógicos	Swap: swapLV de 1204 MB /: rootLV de 8 GB (ext4) /usr: rootLV de 5 GB (ext4) /var: varLV de 4 GB (ext4)
Interfaz gráfica	No
Paquetes	Por defecto
Red	Cliente DHCP

3. Inicie una sesión como bob y visualice los PV, el VG y los LV.

2. Particionado post-instalación

1. Cree un PV /dev/sdc.
2. Agregue en el PV sysVG.
3. Aumente el tamaño del volumen lógico varLV 3 GB.
4. Cree dos PV /dev/sdd y /dev/sde.
5. Cree el VG appVG y añada los dos PV.
6. Cree dos LV: progLV y dataLV.
7. Formatee progLV en ext4 y dataLV en ext3.
8. Monte los dos LV automáticamente al arrancar el servidor.
9. Cree el directorio infos en dataLV.
10. Cree cuatro archivos arch01, arch02, arch03 y arch04 en el directorio infos.

11. Convierta DataLV a ext4 sin perder los datos.

Introducción

Este capítulo tiene como objetivo estudiar la secuencia de arranque del sistema operativo Linux.

Proceso de arranque

Al encender el equipo, el proceso de arranque se realiza en varias etapas.

- La BIOS realiza un POST (Power-On Self Test) para verificar el correcto funcionamiento del equipo.
- El MBR (Master Boot Record), situado en el primer sector del disco de arranque, ejecutará la carga del gestor de arranque (bootloader) GRUB. En cuanto a los sistemas x86 basados en UEFI, montan una partición de sistema EFI que contiene una versión del gestor de arranque GRUB. El gestor de arranque EFI carga y ejecuta GRUB como una aplicación EFI.
- El gestor de arranque GRUB (Grand Unified Bootloader) Legacy (versión 1) o GRUB-PC (versión 2) ofrece en una pantalla de inicio (splash screen) la opción del kernel de Linux a iniciar, por supuesto, si su sistema cuenta con varios. Un retardo de unos pocos segundos nos permite realizar la elección, en caso contrario, el proceso de arranque continúa con los parámetros definidos por defecto en el archivo de configuración de GRUB.
- El kernel de Linux inicializa los dispositivos empleando los controladores, e inicia el proceso kernel kswapd que es el gestor de swap. Luego monta la raíz del sistema de archivos, /.
- Después ejecuta el programa /sbin/init, el cual tendrá el PID 1. Su función es coordinar el final del proceso de arranque y configurar a su vez el entorno del usuario. Se convierte en el padre o abuelo de todos los procesos que se inician automáticamente en el sistema.
- En cuanto a Debian 8, el programa /sbin/init es un enlace simbólico a /lib/systemd/systemd, así como /bin/systemd, por otra parte. Reemplaza al proceso Init System V conservando la compatibilidad con él. Seguimos encontrando los archivos /etc/rc*.d, /etc/rc.local y /etc/init.d. Sin embargo, no encontraremos ningún archivo de configuración /etc/inittab. Los niveles de ejecución (runlevels) han dejado de tener sentido. El valor por defecto es 5 en lugar de 2 como en las versiones anteriores.
- En cuanto a Ubuntu Server 14.04 LTS, este emplea upstart que ya no utiliza el archivo /etc/inittab. La carpeta /etc/init es la ubicación de los archivos de inicialización de upstart. Sin embargo, se conserva la carpeta /etc/init.d por compatibilidad con el sistema Init System V. El archivo /etc/init/rc.sysinit.conf gestiona la ejecución de los comandos de scripts tradicionales que se han añadido manualmente o bien con update-rc.d para los niveles de ejecución tradicionales en los archivos /etc/rc*. /etc/default contiene los archivos de configuración que nos permiten controlar el comportamiento de los

scripts de Init System V tradicionales y las configuraciones de upstart. El nivel de ejecución por defecto es el 2. La próxima versión de LTS, o sea la 16.04, utilizará systemd.

- CentOS utilizaba, hasta su Versión 5, Init System V. La versión 6 ofrecía upstart. En la actualidad la versión 7 usa systemd.

Al parecer, systemd está llamado a convertirse en la herramienta estándar en la mayoría de distribuciones.

GRUB 2

Debian, Ubuntu Server y CentOS utilizan GRUB-PC, es decir, GRUB versión 2. Se convierte poco a poco en el software de arranque estándar de Linux.

1. Identificar la versión de GRUB

Sintaxis

```
grub-install <opción>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-v	--version	Muestra la versión.

Ejemplo con Debian

Mostrar la versión de GRUB:

```
# grub-install -v
grub-install (GRUB) 1.99-27+deb7u1
```

Aunque la versión mostrada sea la 1.99, es una versión 2.

2. Archivos de configuración

El archivo que contiene los parámetros del menú es `/etc/default/grub`. Puede variar ligeramente entre las diferentes distribuciones.

```
# cat /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg

# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=5

GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel
that obtains
# the memory map information from GRUB (GNU Mach, kernel of
FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card
supports via VBE
```

```
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx"
parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

He aquí el significado de estas directivas:

Directiva	Descripción
GRUB_DEFAULT=<valor>	Define el sistema a arrancar por defecto. 0 es el primer elemento del menú.
GRUB_TIMEOUT=<valor>	Arranca el sistema por defecto después del retardo establecido.
GRUB_DISTRIBUTOR=<valor>	Permite identificar la distribución.
GRUB_CMDLINE_LINUX_DEFAULT=<valor>	Se generarán dos entradas para cada elemento del menú: una entrada normal y una en modo de rescate.
GRUB_CMDLINE_LINUX	Permite añadir comandos en las entradas del menú.
GRUB_TERMINAL=console	Desactiva el terminal GUI para un terminal CLI.
GRUB_DISABLE_LINUX_UUID=true	Si esta directiva tiene el valor true entonces Grub-

	mkconfig no generará las entradas de menú utilizando los UUID.
GRUB_DISABLE_RECOVERY=true	Si esta directiva tiene el valor true entonces no se generará ninguna entrada en el menú en modo de rescate.
GRUB_INIT_TUNE=<valores>	Emite un sonido al arrancar.

El directorio /etc/grub.d contiene los scripts de creación del menú. Sus nombres empiezan por una cifra que determina el orden de ejecución:

```
# ls -l
total 72
-rwxr-xr-x 1 root root 9424 dic 23 2015 00_header
-rwxr-xr-x 1 root root 6058 may 23 2015 05_debian_theme
-rwxr-xr-x 1 root root 12261 dic 23 2015 10_linux
-rwxr-xr-x 1 root root 11082 mar 23 2015 20_linux_xen
-rwxr-xr-x 1 root root 11692 dic 23 2015 30_os-prober
-rwxr-xr-x 1 root root 1416 dic 23 2015 30_uefi-firmware
-rwxr-xr-x 1 root root 214 dic 23 2015 40_custom
-rwxr-xr-x 1 root root 216 dic 23 2015 41_custom
-rw-r--r-- 1 root root 483 dic 23 2015 README
```

El archivo /boot/grub/grub.cfg no debe modificarse, ya que se genera de forma automática por el comando grub-mkconfig utilizando los parámetros del archivo /etc/default/grub y plantillas almacenadas en la carpeta /etc/grub.d:

```
# cat grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
```

```
### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
    load_env
fi
...

### BEGIN /etc/grub.d/41_custom ###
if [ -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
fi
### END /etc/grub.d/41_custom ###
```

Si modificamos el archivo `/etc/default/grub` o añadimos un script en la carpeta `/etc/grub.d`, tendremos que usar el comando `update-grub`, `update-grub2` o `grub-mkconfig` para actualizar GRUB en las distribuciones Debian y Ubuntu Server. Para CentOS, debemos utilizar `grub2-mkconfig`.

Sintaxis

```
update-grub
```

O bien :

```
update-grub2
```

Este comando no tiene opciones.

Sintaxis

```
grub-mkconfig <opción>
```

O bien:

```
grub2-mkconfig <opción>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-o <archivo>	--output <archivo>	Permite especificar el nombre del archivo a generar.

Exemple con Debian

update-grub

```
Generating grub configuration file ...
Encontrada imagen de Linux: /boot/vmlinuz-3.13.0-48-generic
Encontrada imagen de memoria inicial: /boot/vmlinuz-3.13.0-48-generic
Encontrada imagen de memoria inicial: /boot/vmlinuz-3.13.0-43-generic
Encontrada imagen de memoria inicial: /boot/initramfs-3.13.0-
43.1.el7.x86_64.img
Found memtest86+ image: /memtest86+.elf
Found memtest86+ image: /memtest86+.bin
hecho
```

O bien:

grub2-mkconfig -o /boot/grub/grub.cfg

```
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.3.0
Found initrd image: /boot/initramfs-4.3.0.img
Found linux image: /boot/vmlinuz-3.10.0-229.20.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-
229.20.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-229.14.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-
229.14.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-
85d58cfe9e95420caf192d072326775e
Found initrd image: /boot/initramfs-0-rescue-
85d58cfe9e95420caf192d072326775e.img
done
```

Antes de actualizar los cambios, podemos probar haciendo una simulación con grub-mkconfig sin opciones. El resultado de la ejecución es un script que se envía a la salida estándar (stdout), es decir la pantalla.

Sintaxis

```
grub-mkconfig
```

O bien:

```
grub2-mkconfig
```

Este comando no tiene opciones.

Exemple con Debian

```
# grub-mkconfig
Generating grub configuration file ...
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
    set have_grubenv=true
    load_env
fi

*** Salida truncada ***

### BEGIN /etc/grub.d/41_custom ###
if [ -f ${config_directory}/custom.cfg ]; then
    source ${config_directory}/custom.cfg
elif [ -z "${config_directory}" -a -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
fi
### END /etc/grub.d/41_custom ###
hecho
```

Guardar y restaurar el MBR

Para efectuar la copia de seguridad del MBR del disco `/dev/sda` en el archivo `mbr.backup`, utilizamos el comando `dd`:

```
# dd if=/dev/sda of=mbr.backup bs=512 count=1
1+0 registros leídos
1+0 registros escritos
512 bytes (512 B) copiados, 0,000370438 s, 1,4 MB/s
```

Para restaurar el MBR del disco `/dev/sda` desde el archivo `mbr.backup`:

```
# dd if=mbr.backup of=/dev/sda bs=512 count=1
1+0 registros leídos
1+0 registros escritos
512 bytes (512 B) copiados, 0,000196174 s, 2,6 MB/s
```

GRUB también puede recuperar el MBR de forma automática invocando `grub-install`.

Sintaxis

```
grub-install <disco>
```

Este comando no tiene opciones.

Ejemplo con Debian

```
# grub-install /dev/sda
Installation finished. No error reported.
```

Debemos reiniciar el equipo para comprobar la secuencia de arranque con el MBR recuperado.

Ejercicio

Deberá utilizar la máquina virtual CentOS 7.

1. Modifique el retardo por defecto del menú GRUB de forma permanente.
2. Reinicie el sistema para verificar.

Introducción

Los diferentes tipos de errores de una aplicación pueden catalogarse así:

- aplicación que no ejecuta,
- aplicación que no responde,
- aplicación con un rendimiento degradado,
- aplicación con un comportamiento inesperado.

Aplicación que no ejecuta

Para diagnosticar una aplicación que falla al arrancar, debemos examinar los siguientes elementos:

- la descripción del paquete DEB o RPM origen de la instalación del programa. En el caso de que el programa haya sido compilado por nosotros, consultamos los parámetros de compilación;
- los archivos de configuración;
- el registro de eventos;
- los filtros de red;
- las dependencias necesarias;
- la ubicación de las librerías;
- la ejecución de la aplicación en modo depuración (debug).

1. Descripción de un paquete instalado

El comando varía de una distribución a otra:

Debian y Ubuntu

Comience por ver la descripción del paquete con el comando `apt-cache show`.

Sintaxis

```
apt-cache show <nombre_del_paquete>
```

Este comando no tiene opciones.

Exemplo con Debian

Mostrar la descripción del programa Apache 2:

```
# apt-cache show apache2
Package: apache2
Version: 2.4.10-10+deb8u4
Installed-Size: 506
Maintainer: Debian Apache Maintainers <debian-
apache@lists.debian.org>
Architecture: amd64
Depends: apache2-mpm-worker (= 2.2.22-13) | apache2-mpm-prefork (=
2.2.22-13) | apache2-mpm-event (= 2.2.22-13) | apache2-mpm-itk (=
```

2.2.22-13), apache2.2-common (= 2.2.22-13)

Description-en: Apache HTTP Server

The Apache HTTP Server Project's goal is to build a secure, efficient and extensible HTTP server as standards-compliant open source software. The result has long been the number one web server on the Internet.

Installing this package results in a full installation, including the configuration files, init scripts and support scripts.

Description-md5: d02426bc360345e5acd45367716dc35c

Homepage: <http://httpd.apache.org/>

Pre-Depends: dpkg (>= 1.17.14)

Recommends: ssl-cert

Section: httpd

Priority: optional

Filename: pool/updates/main/a/apache2/apache2_2.4.10-10+deb8u1_amd64.deb

Size: 205030

MD5sum: 04241cf30710ab7640e4c1b164044f16

SHA1: 27906a59634a9cb7edc98e16aacdec1198f80487

SHA256:

5c98c55ac972cb1ce056c6902be5ce74b7f1afe9d106899d76754d6b5b21495e

Aunque las dependencias pueden visualizarse con `apt-cache show`, el comando `apt-cache depends` realiza también esta operación.

Sintaxis

```
apt-cache depends <nombre_del_paquete>
```

Este comando no tiene opciones.

Ejemplo con Debian

Mostrar las dependencias del programa Apache 2:

```
# apt-cache depends apache2
```

```
apache2
```

```
Depende: lsb-base
```

Depende: procps
Depende: perl
Depende: mime-support
Depende: apache2-bin

CentOS

Con yum info obtendremos la información del paquete en cuestión.

Sintaxis

```
yum info <nombre>
```

Este comando no tiene opciones.

Ejemplo

```
# yum info httpd
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mir01.syntis.net
* extras: mir01.syntis.net
* updates: mir01.syntis.net
Paquetes disponibles
Nombre           : httpd
Arquitectura     : x86_64
Versión          : 2.4.6
Lanzamiento      : 40.el7.centos
Tamaño           : 2.7 M
Repositorio      : base/7/x86_64
Resumen          : Apache HTTP Server
URL              : http://httpd.apache.org/
Licencia         : ASL 2.0
Descripción      : The Apache HTTP Server is a powerful, efficient,
and extensible
                  : web server.
```

También podemos usar el comando rpm para conocer los archivos de configuración o leer la información de un paquete.

Sintaxis

```
rpm <opciones> <nombre>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-q	(ninguna)	Opción de comando.
-c	--configfiles	Enumera los archivos de configuración.
-i	(ninguna)	Muestra información del paquete.

Ejemplos:

Mostrar los archivos de configuración de Apache 2.

```
# rpm -qc httpd
/etc/httpd/conf.d/autoindex.conf
/etc/httpd/conf.d/userdir.conf
/etc/httpd/conf.d/welcome.conf
/etc/httpd/conf.modules.d/00-base.conf
/etc/httpd/conf.modules.d/00-dav.conf
/etc/httpd/conf.modules.d/00-lua.conf
/etc/httpd/conf.modules.d/00-mpm.conf
/etc/httpd/conf.modules.d/00-proxy.conf
/etc/httpd/conf.modules.d/00-systemd.conf
/etc/httpd/conf.modules.d/01-cgi.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/magic
/etc/logrotate.d/httpd
/etc/sysconfig/htcacheclean
/etc/sysconfig/httpd
```

Obtener información del paquete Apache 2.

```
# rpm -qi httpd
Name       : httpd
Version    : 2.4.6
Release    : 31.el7.centos.1
Architecture: x86_64
Install Date: dim. 01 nov. 2015 16:30:58 CET

Group      : System Environment/Daemons
Size       : 9805894
License    : ASL 2.0
Signature  : RSA/SHA256, mar. 25 ago 2015 17:14:33 CEST, Key ID
24c6a8a7f4a80eb5
Source RPM : httpd-2.4.6-31.el7.centos.1.src.rpm
Build Date : lun. 24 ago 2015 20:13:25 CEST
Build Host  : worker1.bsys.centos.org
Relocations : (not relocatable)
Packager    : CentOS BuildSystem <http://bugs.centos.org>
Vendor      : CentOS
URL         : http://httpd.apache.org/
Summary     : Apache HTTP Server
Description :

The Apache HTTP Server is a powerful, efficient, and extensible
web server.
```

2. Archivos de configuración

Algunos programas ofrecen una utilidad que permite comprobar la sintaxis de sus archivos de configuración.

Ejemplo con Debian

Para comprobar la configuración del programa Apache 2, empleamos la utilidad `apachectl`.

He aquí un resultado sin errores.

```
# apachectl -t
```

```
Syntax OK
```

Con un error de sintaxis:

```
# apachectl -t
```

```
Syntax error on line 187 of /etc/apache2/apache2.conf:
```

```
Invalid command 'AccessFileNam', perhaps misspelled or defined by  
a module not included in the server configuration
```

```
Action '-t' failed.
```

```
The Apache error log may have more information.
```

La línea 187 del archivo /etc/apache2/apache2.conf tiene un comando incorrecto. En efecto, la ortografía del comando AccessFileName es errónea: le falta una «e».

Aquí tenemos otra solución para probar:

```
# apache2ctl configtest
```

```
Syntax error on line 187 of /etc/apache2/apache2.conf:
```

```
Invalid command 'AccessFileNam', perhaps misspelled or defined by  
a module not included in the server configuration
```

```
Action 'configtest' failed.
```

```
The Apache error log may have more information.
```

3. Consultar los registros

Por lo general, cuando un programa falla porque no puede acceder a un archivo o a un recurso, escribe una entrada en un registro del sistema o en su propio registro.

En una pantalla desprovista de interfaz gráfica local o remota, podemos usar un multiplexor de terminales en modo texto como GNUscreen o una versión mejorada byobu. Estas aplicaciones no están instaladas por defecto.

De este modo, en una ventana podemos estar probando la aplicación y en otra ventana revisando el registro de eventos en tiempo real con el comando tail.

Instalación en Debian y Ubuntu

Estas distribuciones tienen en sus respectivos almacenes los paquetes del programa screen y byobu. Puestos a elegir, es mejor optar por byobu que es mucho más fácil de usar.

```
# apt-get -y install byobu
```

```
...
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma
automática y ya no son necesarios.
  libasn1-8-heimdal libgssapi3-heimdal libhcrypto4-heimdal
  libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal
  libkrb5-26-heimdal libroken18-heimdal libwind0-heimdal
Utilice «apt-get autoremove» para eliminarlos.
Paquetes sugeridos:
  apport ccze ttf-ubuntu-font-family update-notifier-common vim
wireless-tools
Paquetes recomendados:
  run-one
Se instalarán los siguientes paquetes NUEVOS:
  byobu
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 0 B/171 kB de archivos.
Se utilizarán 662 kB de espacio de disco adicional después de esta
operación.
Preconfigurando paquetes ...
Seleccionando el paquete byobu previamente no seleccionado.
(Leyendo la base de datos ... 248038 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../archives/byobu_5.87-1_all.deb ...
Desempaquetando byobu (5.87-1) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Procesando disparadores para desktop-file-utils (0.22-1) ...
Procesando disparadores para gnome-menus (3.13.3-6) ...
Procesando disparadores para mime-support (3.58) ...
```



```
Procesando disparadores para hicolor-icon-theme (0.13-1) ...
Configurando byobu (5.87-1) ...
```

Instalación en CentOS

byobu no forma parte de los repositorios de CentOS. En cambio, screen sí está presente.

```
# yum -y install screen
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror0.babylon.network
* extras: mir01.syntis.net
* updates: mirror0.babylon.network
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete screen.x86_64 0:4.1.0-0.23.20120314git3c2946.el7_2
debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package Arquitectura Versión                               Repositorio  Tamaño
=====
Instalando:
screen    x86_64          4.1.0-0.19.20120314git3c2946.el7    base    550 k

Resumen de la transacción
=====
Instalar 1 Paquete

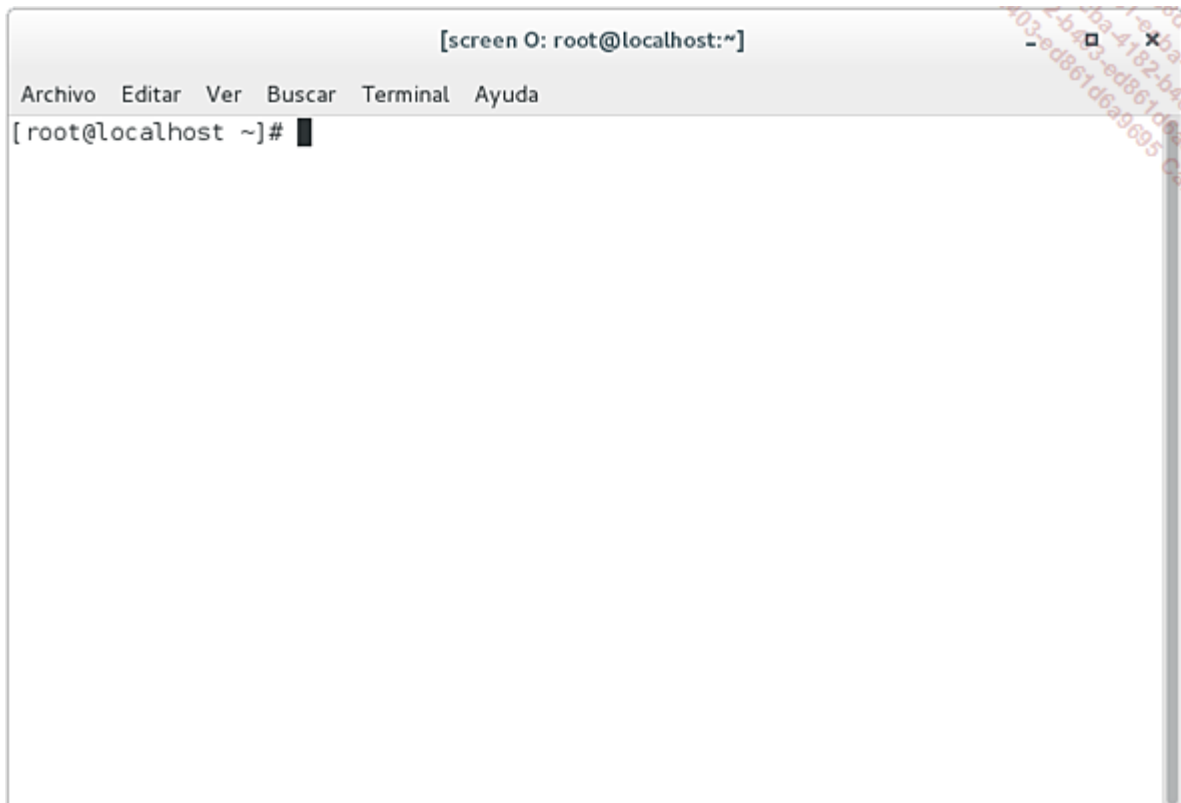
Tamaño total de la descarga: 550 k
Tamaño instalado: 914 k
Downloading Packages:
```

```
screen-4.1.0-0.19.20120314git3c2946.el7.x86_64.rpm | 550 kB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando      : screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64 1/1
  Comprobando    : screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64 1/1

Instalado:
  screen.x86_64 0:4.1.0-0.19.20120314git3c2946.el7

¡Listo!
```

Ya podemos iniciar el programa screen:



Utilice las teclas siguientes para gestionar las ventanas.

Tecla	Descripción
[Ctrl]+A y luego [Ctrl]+C	Crea una ventana adicional.
[Ctrl]+A y luego "	Muestra la lista de ventanas.
[Ctrl]+A y nº de la ventana	Cambia a la ventana seleccionada.
[Ctrl]+D	Cierra la ventana actual.

Si deseamos instalar byobu, es necesario que el paquete gettext esté instalado, que debería ser el caso. En caso contrario lo añadimos con el comando yum.

A continuación debemos buscar el paquete byobu en el sitio <http://rpm.pbone.net> y descargarlo con wget.

```
# wget ftp://mirror.switch.ch/pool/4/mirror/epel/7/x86_64/b/
byobu-5.73-4.el7.noarch.rpm
--2016-05-03 16:11:07--
  ftp://mirror.switch.ch/pool/4/mirror/epel/7/x86_64/b/byobu-5.73-4.
el7.noarch.rpm
      => "byobu-5.73-4.el7.noarch.rpm"
Resolviendo mirror.switch.ch (mirror.switch.ch)... 130.59.10.36,
2001:620:0:8::20
Conectando con mirror.switch.ch
(mirror.switch.ch)[130.59.10.36]:21... conectado.
Identificándose como anonymous ... ¡Dentro!
==> SYST ... hecho.    ==> PWD ... hecho.
==> TYPE I ... hecho.  ==> CWD (1) /pool/4/mirror/epel/7/x86_64/b ...
hecho.
==> SIZE byobu-5.73-4.el7.noarch.rpm ... 158996
==> PASV ... hecho.    ==> RETR byobu-5.73-4.el7.noarch.rpm ... hecho.
Longitud: 158996 (155K) (probablemente)
```

```
100%[=====] 158.996    41,6KB/s    en 3,7s
```

```
2016-05-03 16:11:11 (41,6 KB/s) - «byobu-5.73-4.el7.noarch.rpm»
```

```
guardado [158996]
```

Para instalar el paquete byobu-5.73-4.el7.noarch.rpm es necesaria la dependencia tmux. Esta se encuentra en los repositorios de CentOS:

```
# yum -y install tmux
```

```
Complementos cargados:fastestmirror, langpacks
```

```
Loading mirror speeds from cached hostfile
```

```
* base: mirror0.babylon.network
```

```
* extras: ftp.ciril.fr
```

```
* updates: mirror0.babylon.network
```

```
Resolviendo dependencias
```

```
--> Ejecutando prueba de transacción
```

```
---> Paquete tmux.x86_64 0:1.8-4.el7 debe ser instalado
```

```
--> Resolución de dependencias finalizada
```

```
Dependencias resueltas
```

```
=====
```

Package	Arquitectura	Versión	Repositorio	Tamaño

```
=====
```

```
Instalando:
```

tmux	x86_64	1.8-4.el7	base	243 k

```
Resumen de la transacción
```

```
=====
```

```
Instalar 1 Paquete
```

```
Tamaño total de la descarga: 243 k
```

```
Tamaño instalado: 558 k
```

```
Downloading packages:
```

tmux-1.8-4.el7.x86_64.rpm	243 kB	00:00

```
Running transaction check
```

```
Running transaction test
```

```
Transaction test succeeded
```

```
Running transaction
```

```
Instalando      : tmux-1.8-4.el7.x86_64          1/1
```

```
Comprobando    : tmux-1.8-4.el7.x86_64          1/1
```

```
Instalado:
```

```
tmux.x86_64 0:1.8-4.el7
```

```
¡Listo!
```

Ahora podemos realizar la instalación del paquete byobu con el comando rpm:

```
# rpm -ivh byobu-5.73-4.el7.noarch.rpm
```

```
advertencia:byobu-5.73-4.el7.noarch.rpm: EncabezadoV3 RSA/SHA256
```

```
Signature, ID de clave 352c64e5: NOKEY
```

```
Preparando... ##### [100%]
```

```
Actualizando / instalando...
```

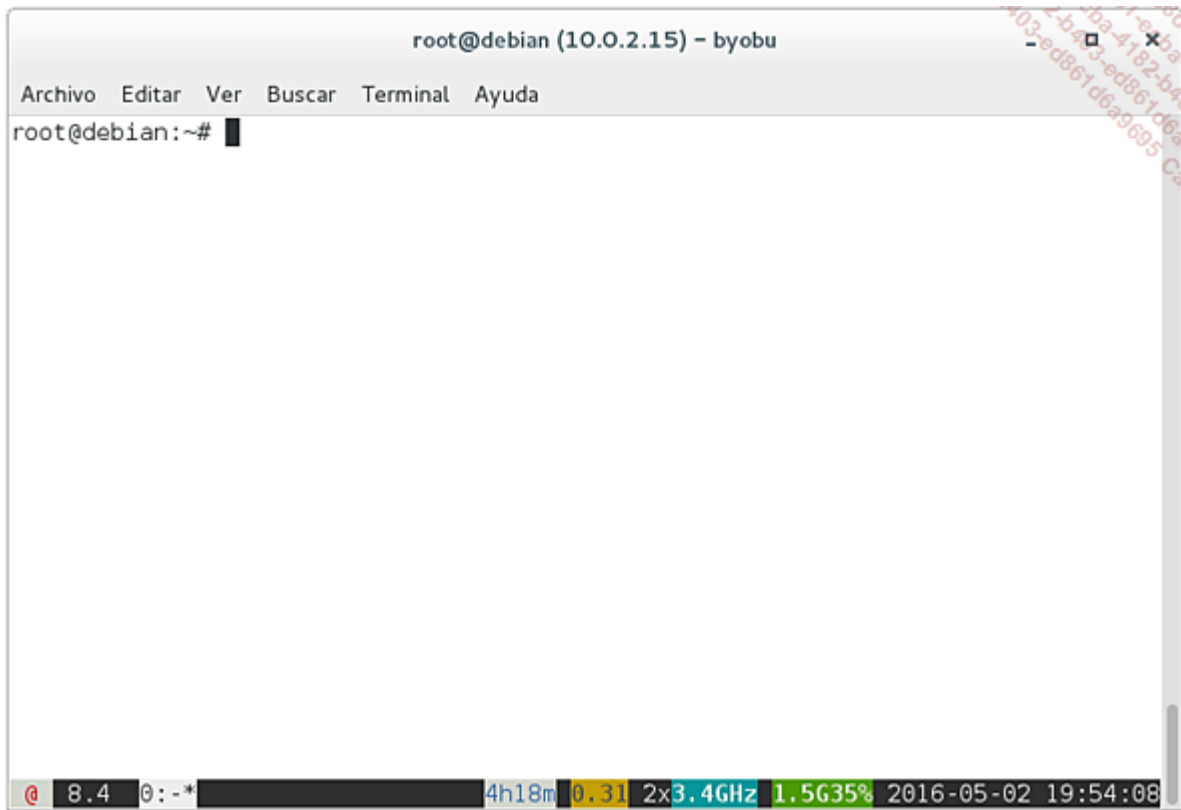
```
1:byobu-5.73-4.el7 ##### [100%]
```

Utilización de byobu

La utilización de byobu permite crear dos ventanas durante un diagnóstico. La primera sirve para recoger las nuevas entradas en un registro y la segunda sirve para ejecutar o manipular la aplicación.

Ejemplo con Debian

Desde un terminal, como usuario root escribimos byobu. Obtendremos esta pantalla:



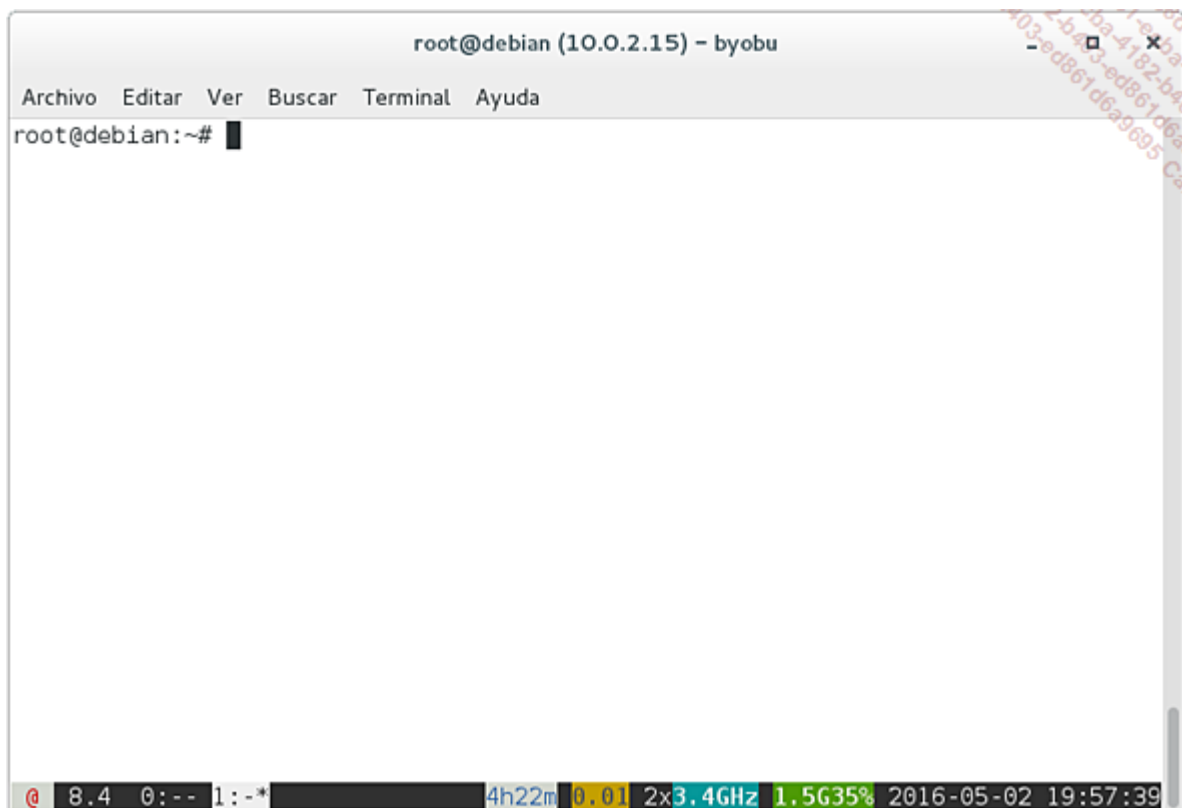
En la barra de estado situada en la parte inferior de la pantalla, veremos que estamos en la primera ventana identificada 0:.

Utilizamos las teclas siguientes para gestionar las ventanas:

<https://dogramcode.com/libros-sistemas>

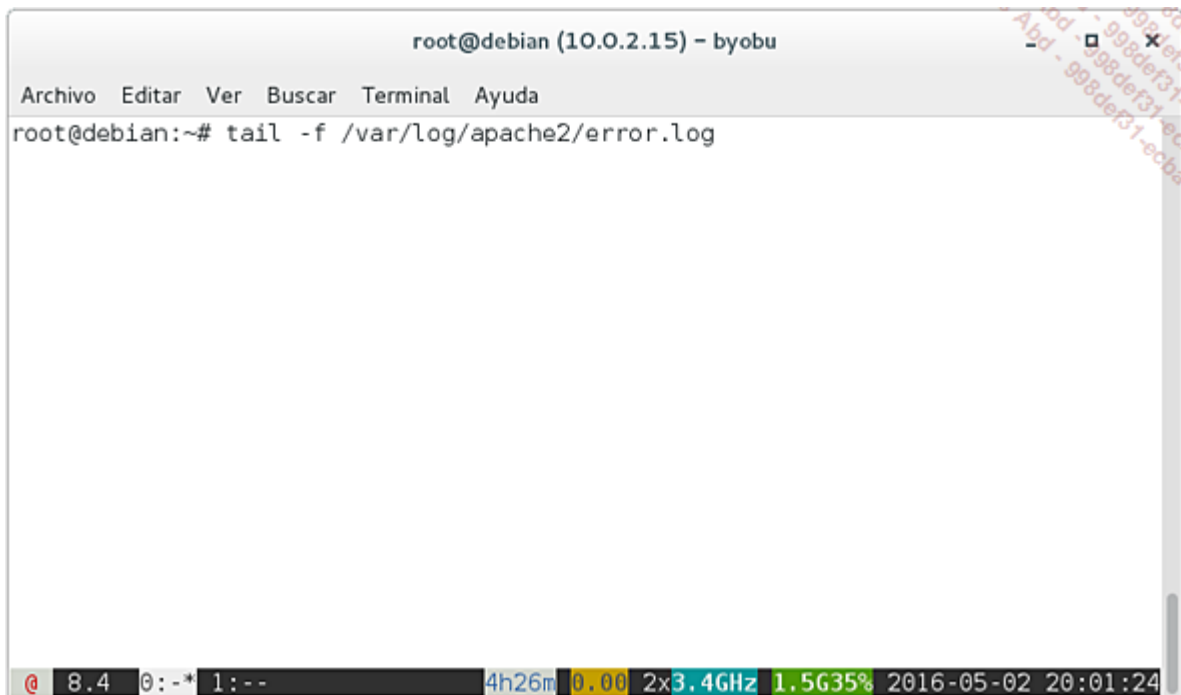
Tecla	Descripción
[F2]	Crea una ventana adicional.
[F3]	Permite desplazarse hacia la ventana de la izquierda.
[F4]	Permite desplazarse hacia la ventana de la derecha.
[F8]	Proporciona un nombre a la ventana.
[F9]	Muestra el menú de configuración.

Para añadir una nueva ventana, pulse la tecla [F2]:



En la ventana 0: escriba `tail -f /var/log/apache2/error.log`:

<https://dogramcode.com/libros-sistemas>

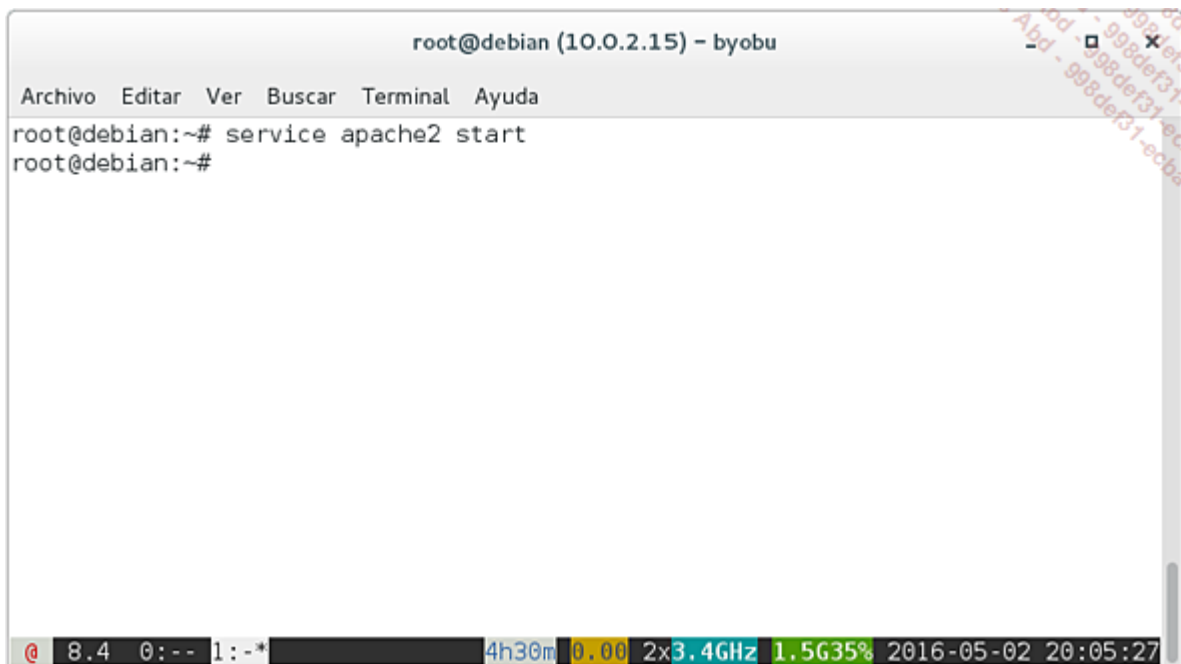


```
root@debian (10.0.2.15) - byobu
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~# tail -f /var/log/apache2/error.log
```

The terminal window shows the command `tail -f /var/log/apache2/error.log` being executed. The window title is `root@debian (10.0.2.15) - byobu`. The menu bar includes `Archivo`, `Editar`, `Ver`, `Buscar`, `Terminal`, and `Ayuda`. The status bar at the bottom displays system information: `@ 8.4 0:-- 1:-- 4h26m 0.00 2x3.4GHz 1.5G35% 2016-05-02 20:01:24`.

Este comando mostrará las entradas del registro `error.log` conforme se vayan creando.

Luego, volvemos a la ventana 1: pulsando [F4] para iniciar el servicio Apache2:



```
root@debian (10.0.2.15) - byobu
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~# service apache2 start
root@debian:~#
```

The terminal window shows the command `service apache2 start` being executed. The window title is `root@debian (10.0.2.15) - byobu`. The menu bar includes `Archivo`, `Editar`, `Ver`, `Buscar`, `Terminal`, and `Ayuda`. The status bar at the bottom displays system information: `@ 8.4 0:-- 1:-- 4h30m 0.00 2x3.4GHz 1.5G35% 2016-05-02 20:05:27`.

Pulsando [F3], consultamos en la ventana 0: la entrada en el registro:


```
root@debian (10.0.2.15) - byobu
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~# tail -f /var/log/apache2/error.log
[Mon May 02 20:04:25.562969 2016] [mpm_event:notice] [pid 16915:tid 140508263868288] AH00489: Apache/2.4.10 (Debian) configured -- resuming normal operations
[Mon May 02 20:04:25.563024 2016] [core:notice] [pid 16915:tid 140508263868288] AH00094: Command line: '/usr/sbin/apache2'
█
```

4. Filtros de red

Es importante verificar si existen reglas de seguridad que podrían bloquear o restringir el funcionamiento del programa.

El comando `netstat` permite ver las conexiones de red, en particular los puertos TCP o UDP que estén en escucha y su estado.

Sintaxis

```
netstat <opciones>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-t	--tcp	Muestra el protocolo TCP.
-u	--udp	Muestra el protocolo UDP.
-l	--listening	Muestra los sockets del servidor en escucha.

-p	--programs	Muestra el nombre y el PID de los procesos propietarios.
-n	--numeric	Muestra las direcciones en formato numérico.

Ejemplo

Muestra el puerto de escucha del programa Apache 2:

```
# netstat -tulpn |grep apache2
tcp6      0  0 :::80      :::*       LISTEN    0      15732      4991/apache2
```

La escucha se hace sobre la dirección IPv4 10.0.2.15 en el puerto TCP 80.

El comando ss también permite visualizar los puertos en escucha:

Sintaxis

```
ss <opciones>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-t	--tcp	Muestra el protocolo TCP.
-u	--udp	Muestra el protocolo UDP.
-l	--listening	Muestra los sockets del servidor en escucha.
-p	--processes	Muestra el nombre y el PID de los procesos propietarios.
-n	--numeric	Muestra las direcciones en formato numérico.

Ejemplo

Muestra el puerto de escucha del programa Apache 2:

```
# ss -tulpn |grep apache2
tcp    LISTEN    0      0      :::80      :::*      users:
(("apache2",pid=4995,fd=4),("apache2",pid=4994,fd=4),
("apache2",pid=4991,fd=4))
```

Observe la configuración del firewall iptables.

Sintaxis

```
iptables <opciones>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-v	--verbose	Modo largo (verbose).
-L	--list	Enumera las reglas definidas.

Ejemplo

```
# iptables -vL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source               destination
```

5. Dependencias del programa

El programa puede recurrir a funciones almacenadas en las librerías. El comando ldd (List Dynamic Dependencies) muestra una lista de librerías compartidas que son necesarias para un programa o una librería.

Sintaxis

```
ldd <archivo>
```

Opción útil:

Opción		Descripción
corta	larga	
-v	--verbose	Modo largo (verbose).

Ejemplo con Debian

Mostrar las bibliotecas necesarias como dependencia para el programa Apache 2:

```
# ldd $(which apache2)
linux-vdso.so.1 (0x00007ffe6bf73000)
    libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3
(0x00007f9e7fa43000)
    libaprutil-1.so.0 => /usr/lib/x86_64-linux-gnu/libaprutil-1.so.0
(0x00007f9e7f81a000)
    libapr-1.so.0 => /usr/lib/x86_64-linux-gnu/libapr-1.so.0
(0x00007f9e7f5e4000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0
(0x00007f9e7f3c7000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f9e7f01e000)
    libuuid.so.1 => /lib/x86_64-linux-gnu/libuuid.so.1
(0x00007f9e7ee18000)
    librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1
(0x00007f9e7ec10000)
    libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1
(0x00007f9e7e9d9000)
```

```
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2
(0x00007f9e7e7d4000)

libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1
(0x00007f9e7e5ab000)

/lib64/ld-linux-x86-64.so.2 (0x0000560e1d3b1000)
```

A su vez, una librería compartida puede tener una dependencia. Y así sucesivamente... He aquí el ejemplo de libc.so.6:

```
# ldd /lib/x86_64-linux-gnu/libc.so.6
/lib64/ld-linux-x86-64.so.2 (0x000055e0dd780000)
linux-vdso.so.1 (0x00007ffc8aec1000)
```

La opción -v muestra la información de manera más detallada:

```
ldd -v /lib/x86_64-linux-gnu/libc.so.6
/lib64/ld-linux-x86-64.so.2 (0x0000564e91110000)
linux-vdso.so.1 (0x00007ffc54da0000)

Version information:
/lib/x86_64-linux-gnu/libc.so.6:
    ld-linux-x86-64.so.2 (GLIBC_2.3) => /lib64/ld-linux-x86-64.so.2
    ld-linux-x86-64.so.2 (GLIBC_PRIVATE) => /lib64/ld-linux-x86-64.so.2
```

6. Ubicación de las librerías

Cuando una librería no se encuentra en su ubicación estándar, aparecerá ausente para el cargador. Habrá que añadir su ruta a la variable de entorno LD_LIBRARY_PATH para que se la pueda tener en cuenta:

```
$ export LD_LIBRARY_PATH=/home/bob/lib
```

Este método se utiliza generalmente para la depuración de un programa.

Si se desea agregar definitivamente una ubicación de librerías, es preferible utilizar el archivo /etc/ld.so.conf:

```
# cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf
```

Contiene una directiva que incluye todos los archivos del directorio /etc/ld.so.conf.d cuya extensión sea .conf.

La carpeta /etc/ld.so.conf.d/ contiene una lista de archivos que varía según la distribución:

```
# ls -l /etc/ld.so.conf.d/
total 12
-rw-r--r-- 1 root root 38 oct.  5 2014 fakeroot-x86_64-linux-gnu.conf
-rw-r--r-- 1 root root 44 ago   9 2009 libc.conf
-rw-r--r-- 1 root root 68 abr  15 2015 x86_64-linux-gnu.conf
```

El comando ldconfig crea los enlaces y pone en caché las librerías compartidas más recientes que se encuentra en las rutas indicadas por la línea de comandos en el archivo /etc/ld.so.conf y en los archivos /lib y /usr/lib.

Sintaxis

```
ldconfig <opción>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-v	--verbose	Modo largo (verbose).
-p	--print-cache	Muestra los archivos y librerías registradas en la caché.

Ejemplo con Debian

Visualizar el contenido de todos los archivos de las librerías compartidas:

```
# ldconfig -v
...
/sbin/ldconfig.real: Se ha dado la ruta `/lib/x86_64-linux-gnu' más de
una vez
/sbin/ldconfig.real: Se ha dado la ruta `/usr/lib/x86_64-linux-gnu' más
de una vez
/usr/lib/x86_64-linux-gnu/libfakeroot:
    libfakeroot-0.so -> libfakeroot-tcp.so

/usr/local/lib:
```

```
/lib/x86_64-linux-gnu:
    libnss_mdns4_minimal.so.2 -> libnss_mdns4_minimal.so.2
    liblvm2cmd.so.2.02 -> liblvm2cmd.so.2.02
    libmemusage.so -> libmemusage.so
    libresolv.so.2 -> libresolv-2.19.so
*** Salida truncada ***
```

Mostrar el contenido de la caché:

```
# ldconfig -p
...
1467 bibliotecas se encontraron en la caché `/etc/ld.so.cache'
    libzipwrap-0.so.13 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzipwrap-0.so.13
    libzipmmapped-0.so.13 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzipmmapped-0.so.13
    libzipfseeko-0.so.13 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzipfseeko-0.so.13
    libzip-0.so.13 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzip-0.so.13
    libzvbi.so.0 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzvbi.so.0
    libzvbi-chains.so.0 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzvbi-chains.so.0
    libzip.so.2 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/libzip.so.2
    libzephyr.so.4 (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/
libzephyr.so.4      libzeitgeist-2.0.so.0 (libc6,x86-64) => /usr/lib/
x86_64-linux-gnu/libzeitgeist-2.0.so.0
    libzbar.so.0 (libc6,x86-64) => /usr/lib/libzbar.so.0
    libzapojit-0.0.so.0 (libc6,x86-64) => /usr/lib/libzapojit-0.0.so.0
    libz.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libz.so.1

    libz.so (libc6,x86-64) => /usr/lib/x86_64-linux-gnu/libz.so
```

*** Salida truncada ***

El comando `strace` es una herramienta de depuración para monitorizar las llamadas al sistema (system calls) utilizadas por un programa así como todas las señales que recibe.

Sintaxis

```
strace <programa>
```

Este comando no tiene opciones.

Instalación en Debian y Ubuntu

```
# apt-get install strace
```

```
Leyendo lista de paquetes... Hecho
```

```
Creando árbol de dependencias
```

```
Leyendo la información de estado... Hecho
```

```
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
```

```
libasn1-8-heimdal libgssapi3-heimdal libhcrypto4-heimdal
```

```
libheimbase1-heimdal libheimt1m0-heimdal libhx509-5-heimdal
```

```
libkrb5-26-heimdal libroken18-heimdal libwind0-heimdal
```

```
Utilice «apt-get autoremove» para eliminarlos.
```

```
Se instalarán los siguientes paquetes NUEVOS:
```

```
strace
```

```
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

```
Se necesita descargar 274 kB de archivos.
```

```
Se utilizarán 1.014 kB de espacio de disco adicional después de esta operación. Des:1 http://ftp.es.debian.org/debian/jessie/main strace amd64 4.9-2 [274 kB]
```

```
Descargados 274 kB en 1s (179 kB/s)
```

```
Seleccionando el paquete strace previamente no seleccionado.
```

```
(Leyendo la base de datos ... 248209 ficheros o directorios instalados actualmente.)
```

```
Preparando para desempaquetar .../strace_4.9-2_amd64.deb ...
```

```
Desempaquetando strace (4.9-2) ...
```


Procesando disparadores para man-db (2.7.0.2-5) ...

Configurando strace (4.9-2) ...

Instalación en CentOS

Por defecto, strace está instalado en CentOS. Sin embargo, si esta herramienta no se encuentra instalada en nuestra configuración, podemos añadirla con yum.

Utilización de strace

Ejemplo con Debian

Mostrar las llamadas al sistema utilizadas y las señales recibidas por el programa Apache 2:

```
# strace apache2
...
execve("/usr/sbin/apache2", ["apache2"], [/ * 17 vars */]) = 0
brk(0)                                = 0x55d4c287e000
access("/etc/ld.so.nohwcap", F_OK)     = -1 ENOENT (No such file or
directory)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f936bcd5000
access("/etc/ld.so.preload", R_OK)     = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=37895, ...}) = 0
mmap(NULL, 37895, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f936bccb000
close(3)                               = 0
access("/etc/ld.so.nohwcap", F_OK)     = -1 ENOENT (No such file or
directory)
open("/lib/x86_64-linux-gnu/libpcres.so.3", O_RDONLY|O_CLOEXEC) = 3
*** Salida truncada ***
```

7. Modo de depuración

El programa en cuestión puede incorporar un modo de depuración.

<https://dogramcode.com/libros-sistemas>

Ejemplo con Debian

El archivo de configuración `/etc/apache2/apache2.conf` de Apache 2 tiene una directiva `LogLevel` que define el nivel de detalle de los mensajes que deben guardarse en los registros de error. Cambiamos el valor existente (que es generalmente `warn`) por `debug`:

```
...  
# less /etc/apache2/apache2.conf  
*** Salida truncada ***  
  
#  
# LogLevel: Control the severity of messages logged to the error_log.  
# Available values: trace8, ..., trace1, debug, info, notice, warn,  
# error, crit, alert, emerg.  
# It is also possible to configure the log level for particular modules,  
# e.g. "LogLevel info ssl:warn"  
#  
LogLevel warn  
*** Salida truncada ***
```

Aplicación que no responde

1. Modo depuración

Si una aplicación no responde, puede estar esperando por ejemplo una conexión de red. Podemos vincular el comando `strace` a este proceso.

Sintaxis

```
strace <programa>
```

Opción necesaria:

Opción corta	Descripción
<code>-p <pid></code>	Traza el proceso mencionado mediante su PID.

Ejemplo con Debian

Para obtener la traza del proceso de Apache 2 es preciso conocer su PID:

```
# pidof apache2
4995 4994 4991
```

O bien:

```
# ps ax | grep apache2 | grep -v grep
4991 ?        Ss      0:00 /usr/sbin/apache2 -k start
4994 ?        Sl      0:00 /usr/sbin/apache2 -k start
4995 ?        Sl      0:00 /usr/sbin/apache2 -k start
```

Apache devuelve varios PID. Tomamos el último valor:

```
# strace -p 4991
Process 4991 attached
select(0, NULL, NULL, NULL, {0, 549301}) = 0 (Timeout)
wait4(-1, 0x7ffef5b4643c, WNOHANG|WSTOPPED, NULL) = 0
select(0, NULL, NULL, NULL, {1, 0})      = 0 (Timeout)
wait4(-1, 0x7ffef5b4643c, WNOHANG|WSTOPPED, NULL) = 0
select(0, NULL, NULL, NULL, {1, 0})      = 0 (Timeout)

wait4(-1, 0x7ffef5b4643c, WNOHANG|WSTOPPED, NULL) = 0
*** Salida truncada ***
```

Pulsamos [CTRL]+C para detener el proceso.

2. Terminar un proceso

Podemos solicitar al sistema operativo que termine un proceso que presente un comportamiento anormal o que se haya colgado indicando su identificador de proceso (PID) o bien su nombre.

a. Buscar un PID

Podemos buscar el PID de una aplicación que no responde empleando los siguientes comandos:

- `pidof`
- `pgrep`
- `ps`

`pidof`

Este comando muestra el o los PID del programa invocado.

Sintaxis

```
pidof <opción> <nombre_del_programa>
```

Opción útil:

Opción corta	Descripción
-s	Devuelve un único PID.

Ejemplo

Conocer los PID del programa ssh:

```
# pidof sshd
3610 3605 3423
```

`pgrep`

El comando `pgrep` muestra los identificadores de los procesos.

Sintaxis

```
pgrep <opciones> <nombre_del_programa>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-l	--list-name	Muestra el proceso con su PID.
-u <usuario>	--euid <usuario>	Muestra los procesos del UID solicitado.

Ejemplo

Conocer los PID del programa sshd:

```
# pgrep -l sshd
3423 sshd
3605 sshd
3610 sshd
```

Conocer el PID de los programas cuyo propietario es root:

```
# pgrep -lu root
1790 systemd
1791 (sd-pam)
1793 sshd
1794 bash
```

ps

ps permite mostrar el estado de los procesos en curso.

Sintaxis

```
ps <opciones>
```

Opciones necesarias:

Opción corta	Descripción
-a	Muestra todos los procesos.
-u	Muestra el nombre de usuario y la hora de arranque.
-x	Muestra también los procesos que no tienen un terminal de control.

Ejemplo

Conocer solamente el PID del programa sshd:

```
# ps aux |grep sshd
root      3423  0.0  0.2  49848  1200 ?        Ss   11:10
0:00 /usr/sbin/sshd
root      3605  0.0  0.7  92112  3960 ?        Ss   11:10  0:00
sshd: bob [priv]
bob       3610  0.0  0.3  92112  1904 ?        S    11:10  0:00
sshd: bob@pts/0
root      3847  0.0  0.1   7848   884 pts/0    S+   11:22  0:00
grep sshd
```

Podemos añadir `grep -v grep` para excluir el proceso `grep sshd`.

```
# ps aux |grep sshd |grep -v grep
root      3423  0.0  0.2  49848  1200 ?        Ss   11:10
0:00 /usr/sbin/sshd
root      3605  0.0  0.7  92112  3960 ?        Ss   11:10  0:00
sshd: bob [priv]
bob       3610  0.0  0.3  92112  1904 ?        S    11:10  0:00
sshd: bob@pts/0
```

b. Herramientas para terminar un proceso

Si un programa no responde en absoluto, será necesario enviar una señal al sistema operativo para solicitar que termine su ejecución. Para llevar a cabo esta operación disponemos de varios comandos:

- kill
- pkill
- killall
- xkill

kill

kill permite enviar distintos tipos de señales.

Sintaxis

```
kill <señal> <pid>
```

Las señales útiles:

Señal	Descripción
-2 (SIGINT)	Interrumpe el proceso. [Ctrl]+C utiliza esta señal.
-15 (SIGTERM)	Termina el proceso.
-9 (SIGKILL)	A diferencia de SIGINT y SIGTERM, esta señal no puede ser ignorada por el proceso.

Para conocer la lista de todas las señales disponibles:

```
# kill -l
```

1) SIGHUP	2) SIGINT	3) SIGQUIT
4) SIGILL	5) SIGTRAP	6) SIGABRT
7) SIGBUS	8) SIGFPE	9) SIGKILL
10) SIGUSR1	11) SIGSEGV	12) SIGUSR2
13) SIGPIPE	14) SIGALRM	15) SIGTERM
16) SIGSTKFLT	17) SIGCHLD	18) SIGCONT
19) SIGSTOP	20) SIGTSTP	21) SIGTTIN
22) SIGTTOU	23) SIGURG	24) SIGXCPU

25) SIGXFSZ	26) SIGVTALRM	27) SIGPROF
28) SIGWINCH	29) SIGIO	30) SIGPWR
31) SIGSYS	34) SIGRTMIN	35) SIGRTMIN+1
36) SIGRTMIN+2	37) SIGRTMIN+3	38) SIGRTMIN+4
39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7
42) SIGRTMIN+8	43) SIGRTMIN+9	44) SIGRTMIN+10
45) SIGRTMIN+11	46) SIGRTMIN+12	47) SIGRTMIN+13
48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14
51) SIGRTMAX-13	52) SIGRTMAX-12	53) SIGRTMAX-11
54) SIGRTMAX-10	55) SIGRTMAX-9	56) SIGRTMAX-8
57) SIGRTMAX-7	58) SIGRTMAX-6	59) SIGRTMAX-5
60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX	

Ejemplo

Para terminar el programa Apache 2 de forma adecuada, debemos utilizar la señal SIGTERM:

```
# kill -15 $(pidof apache2)
```

Cabe señalar que el valor por defecto es SIGTERM. Por lo tanto, también podemos escribir lo siguiente:

```
# kill -15 $(pidof apache2)
```

Si el comando SIGTERM fracasa, será necesario enviar la señal SIGKILL.

```
# kill -9 $(pidof apache2)
```

pgrep

pgrep también envía una señal a los procesos en función de su nombre.

Sintaxis

```
pgrep <opciones> <nombre_del_programa>
```


Opciones útiles:

Opción		Descripción
corta	larga	
-l	--list-name	Muestra el proceso con su PID.
-u <usuario>	--euid <usuario>	Muestra los procesos del UID solicitado.
-P	--parent	Selecciona los procesos del PID padre (PPID).

Ejemplo

Mostrar los procesos de Apache 2 para comprobar si está en ejecución:

```
# pidof apache2
4687 4686 4684 4681
```

Terminar los procesos asociados a Apache 2:

```
# pkill apache2
```

O bien:

```
# pkill --signal 15 vi
```

La señal por defecto es SIGTERM (-15).

No necesitamos conocer el PID del proceso, a diferencia de con el comando kill. Sin embargo, todos las instancias de vi finalizan.

Comprobemos el resultado:

```
# pidof apache2
```

killall

killall termina un proceso invocando su nombre. Todas las instancias del proceso invocado finalizarán. Por defecto, el comando envía la señal SIGTERM (-15).

Sintaxis

```
killall <opciones> <procesos>
```

Opciones útiles:

Opción		Descripción	Ejemplo
corta	larga		
-l	--list	Enumera las señales disponibles.	Mostrar los procesos de Apache 2 para saber si está en ejecución: # pidof apache2 4687 4686 4684 4681
-s	--signal	Envía una señal específica	
-u	--user	Termina los procesos del usuario solicitado.	Termine con la señal SIGKILL (-9) los procesos asociados a Apache 2.

```
# killall -s 9 apache2
```

No necesitamos conocer el PID del proceso, a diferencia de con el comando kill.

Comprobemos el resultado:

```
# pidof apache2
```

xkill

Esta herramienta es útil cuando contamos con un entorno gráfico. xkill solicita al servidor X cerrar la conexión con el cliente X. Permite terminar un proceso colgado en general que se ejecuta en una ventana en la pantalla del escritorio.

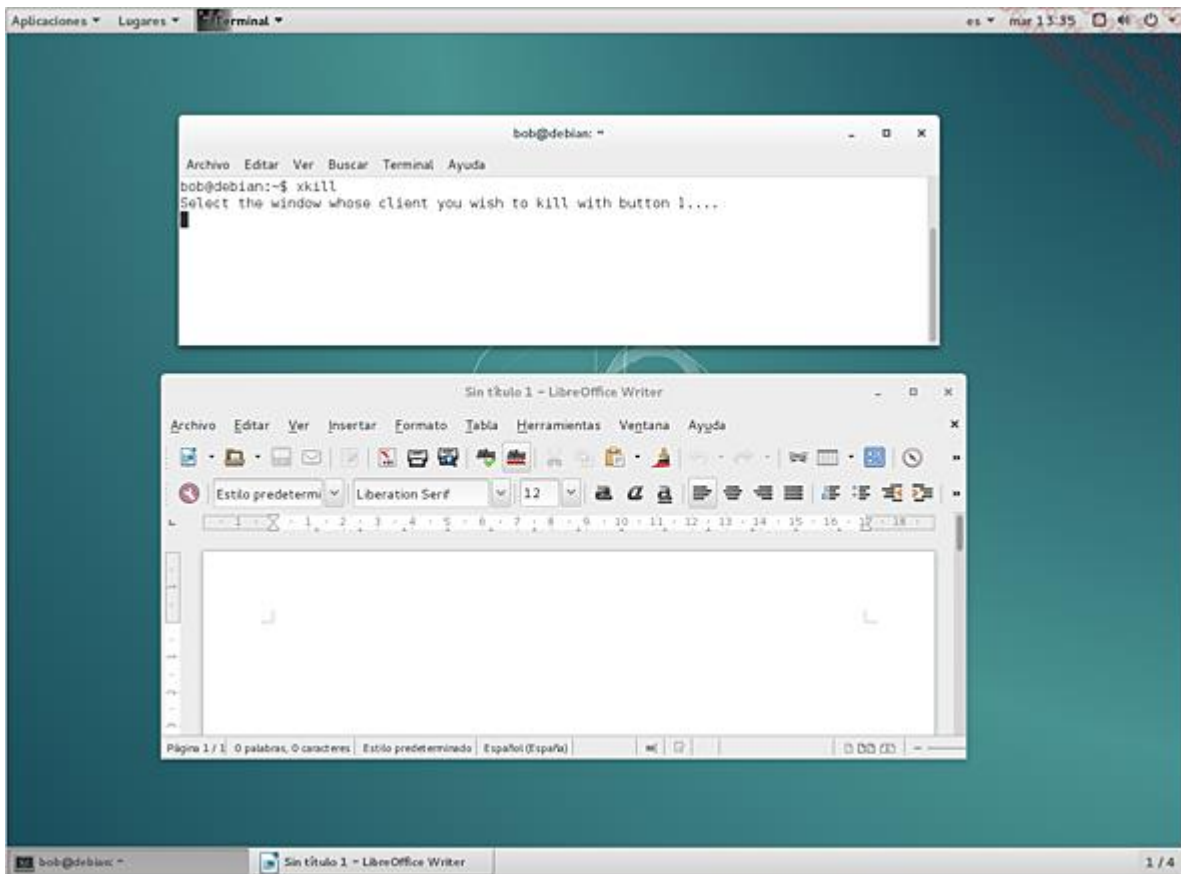
Sintaxis

```
xkill
```

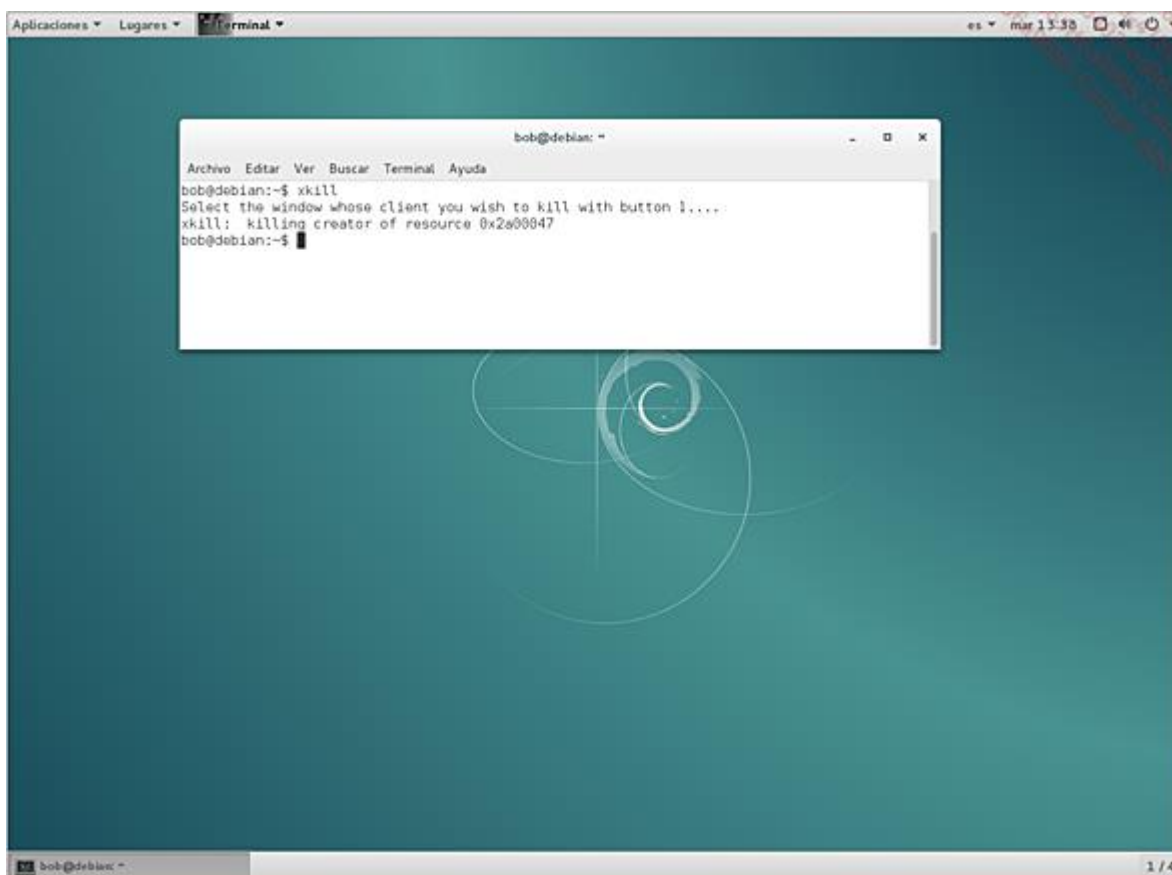
Este comando no tiene opciones.

Ejemplo

Tenemos una aplicación como LibreOffice en el escritorio. Para utilizar xkill, abrimos un terminal sin necesidad de ser root y escribimos xkill:



Desplazamos el ratón sobre la ventana de LibreOffice y hacemos clic:



La ventana sobre la que hemos hecho clic desaparece. El o los procesos en cuestión se terminan.

Rendimiento degradado

Si la aplicación presenta un rendimiento degradado, es sobre todo útil conocer su funcionamiento normal. ¿Existe en realidad una razón legítima para este cambio en el rendimiento? Por otra parte, podría existir una actividad ilícita o maliciosa oculta.

En la siguiente tabla, encontraremos posibles razones para este fallo:

Causa	Solución
La CPU no es suficiente.	Poner una CPU más potente.
La CPU está muy solicitada por los procesos.	Repartir las aplicaciones entre otras CPU o mover las aplicaciones.

El uso de la aplicación ha aumentado considerablemente, por lo que requiere de más recursos.

Optimizar (tuning) la configuración de la aplicación.



El capítulo Análisis del rendimiento - Procesador ofrece herramientas para diagnosticar la actividad de la CPU.

Comportamientos inesperados

Al igual que para un rendimiento degradado, es útil conocer el rendimiento normal de la aplicación correspondiente.

Si la aplicación no presenta un funcionamiento normal, sino por el contrario se comporta de manera extraña, debemos identificar la causa legítima para este cambio de rendimiento. ¿Se produjo el problema después de una actualización? ¿Y cuáles son los elementos impactados?

1. Eliminar los zombis

Un proceso zombi es un proceso que ha sido puesto en marcha por un proceso padre y al que no le notifica (al padre) cuando termina. El sistema limpia la memoria y los recursos utilizados por el proceso zombi. La tabla de procesos mantiene sin embargo constancia del mismo hasta el momento en que el proceso padre solicita al sistema operativo el estado de su proceso hijo. Solo entonces el proceso zombi desaparece.

Tener demasiados procesos zombis provoca un enorme despilfarro de recursos y puede llevar a no poder iniciar nuevos programas, ya que el número de procesos está limitado en el sistema.

Podemos determinar el número máximo de procesos, que suele ser 32768, de esta forma:

```
# cat /proc/sys/kernel/pid_max  
32768
```

Para evitar este tipo de situación, los procesos deben ser escritos para no crear zombis y por extensión no es normal encontrarlos.

Para eliminar un zombi debemos terminar la ejecución del padre. Lo que puede generar la parada de una aplicación necesaria para la explotación.

Para comprobar el estado de los procesos, utilizamos el comando ps.

Sintaxis

```
ps <opciones>
```

Opciones necesarias:

Opción corta	Descripción
-a	Muestra todos los procesos.
-u	Muestra el usuario propietario del proceso.
-x	Muestra también los procesos que no tienen un terminal de control.

Ejemplo

```
...  
# ps aux | grep Z  
USER PID %CPU %MEM VSZ   RSS TTY      STAT START  TIME COMMAND  
bob  7294 0.0  0.0  0      0 pts/9      Z   20:47  0:00 [08_zombi] <defunct>  
bob  7282 0.0  0.0 11812 2316 pts/9  S+  20:48  0:00 grep --colour=auto Z
```

La columna STAT muestra una Z si el proceso se encuentra en estado zombi. Es el caso del proceso que tiene el PID 7294. Para conocer el proceso padre, escribimos:

```
# pstree -p -s 7294  
nit(1)---mdm(1354)---mdm(1381)---init(2525)---yakuake(3509)---bash  
(3682)---08_zombi(7293)---08_zombi(7294)
```

El proceso padre es el programa 08_zombi cuyo PID es 7293. Para matar este programa, utilizamos el comando kill.

```
# kill -9 7293
```

2. Fuga de memoria

Una fuga de memoria es una ocupación creciente y a menudo no controlada de la memoria RAM, provocada por un programa. Esta puede interrumpirse de forma anormal e inesperada sin proporcionar la menor pista de su mal funcionamiento.

Los lenguajes de alto nivel tienen un componente llamado garbage collector que permite reducir el riesgo de fugas de memoria.

El software libre valgrind escrito por Julian Seward bajo licencia GPL versión 2 permite depurar y también identificar las fugas de memoria. Esta herramienta está disponible en los repositorios de Debian, Ubuntu Server y CentOS.

Instalación en Debian y Ubuntu

```
# apt-get -y install valgrind
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  gdb gdbserver libc6-dbg libpython2.7 valgrind-dbg
Paquetes sugeridos:
  gdb-doc valgrind-mpi kcachegrind alleyoop valkyrie
Paquetes recomendados:
  libc-dbg
Se instalarán los siguientes paquetes NUEVOS:
  gdb gdbserver libc6-dbg valgrind valgrind-dbg
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 71,4 MB de archivos.
Se utilizarán 371 MB de espacio de disco adicional después de esta
operación.
Des:1 http://ftp.es.debian.org/debian/ jessie/main gdb amd64
7.7.1+dfsg-5 [2.293 kB]
Des:2 http://ftp.es.debian.org/debian/ jessie/main gdbserver amd64
7.7.1+dfsg-5 [244 kB]
Des:3 http://ftp.es.debian.org/debian/ jessie/main libc6-dbg amd64
2.19-18+deb8u4 [3.430 kB]
Des:4 http://ftp.es.debian.org/debian/ jessie/main valgrind amd64
1:3.10.0-4 [16,7 MB]
Des:5 http://ftp.es.debian.org/debian/ jessie/main valgrind-dbg amd64
1:3.10.0- 4 [48,8 MB]
Descargados 71,4 MB en 20s (3.546 kB/s)
Seleccionando el paquete gdb previamente no seleccionado.
(Leyendo la base de datos ... 247557 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../gdb_7.7.1+dfsg-5_amd64.deb ...
```

```

Desempaquetando gdb (7.7.1+dfsg-5) ...
Seleccionando el paquete gdbserver previamente no seleccionado.
Preparando para desempaquetar .../gdbserver_7.7.1+dfsg-5_amd64.deb ...
Desempaquetando gdbserver (7.7.1+dfsg-5) ...
Seleccionando el paquete libc6-dbg:amd64 previamente no seleccionado.
Preparando para desempaquetar .../libc6-dbg_2.19-18+deb8u4_
amd64.deb ...
Desempaquetando libc6-dbg:amd64 (2.19-18+deb8u4) ...
Seleccionando el paquete valgrind previamente no seleccionado.
Preparando para desempaquetar .../valgrind_1%3a3.10.0-4_amd64.deb ...
Desempaquetando valgrind (1:3.10.0-4) ...
Seleccionando el paquete valgrind-dbg previamente no seleccionado.
Preparando para desempaquetar .../valgrind-dbg_1%3a3.10.0-4_
amd64.deb ...
Desempaquetando valgrind-dbg (1:3.10.0-4) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Configurando gdb (7.7.1+dfsg-5) ...
Configurando gdbserver (7.7.1+dfsg-5) ...
Configurando libc6-dbg:amd64 (2.19-18+deb8u4) ...
Configurando valgrind (1:3.10.0-4) ...
Configurando valgrind-dbg (1:3.10.0-4) ...

```

Instalación en CentOS

```

# yum -y install valgrind
Complementos cargados:fastestmirror, langpacks
base | 3.6 kB 00:00
extras | 3.4 kB 00:00
updates | 3.4 kB 00:00
updates/7/x86_64/primary_db | 4.1 MB 00:01
Loading mirror speeds from cached hostfile
* base: centos.mirrors.ovh.net

* extras: centos.mirrors.ovh.net

```



```

* updates: mirror.ibcp.fr
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete valgrind.x86_64 1:3.10.0-16.el7 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas
=====
Package      Arquitectura  Versión          Repositorio  Tamaño
=====
Instalando:
valgrind     x86_64        1:3.10.0-16.el7  base         15 M

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 15 M
Tamaño instalado: 43 M
Downloading packages:
valgrind-3.10.0-16.el7.x86_64.rpm | 15 MB 00:06
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando      : 1:valgrind-3.10.0-16.el7.x86_64          1/1
  Comprobando    : 1:valgrind-3.10.0-16.el7.x86_64          1/1

Instalado:
valgrind.x86_64 1:3.10.0-16.el7

¡Listo!

```

Ejemplo

Análisis del programa Apache 2:

```
# valgrind --tool=memcheck apache2
==3163== Memcheck, a memory error detector
==3163== Copyright (C) 2002-2011, and GNU GPL'd, by Julian Seward
et al.
==3163== Using Valgrind-3.10.0 and LibVEX; rerun with -h for
copyright info
==3163== Command: apache2
==3163==
==3163== HEAP SUMMARY:
==3163==      in use at exit: 4,325 bytes in 11 blocks
==3163==    total heap usage: 167 allocs, 156 frees, 174,546 bytes
allocated
==3163==
==3163== LEAK SUMMARY:
==3163==    definitely lost: 0 bytes in 0 blocks
==3163==    indirectly lost: 0 bytes in 0 blocks
==3163==    possibly lost: 0 bytes in 0 blocks
==3163==    still reachable: 4,325 bytes in 11 blocks
==3163==          suppressed: 0 bytes in 0 blocks
==3163== Rerun with --leak-check=full to see details of leaked memory
==3163==
==3163== For counts of detected and suppressed errors, rerun with: -v
==3163== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Aquí se aprecia que los valores no devuelven anomalías:

```
==3163==    definitely lost: 0 bytes in 0 blocks
```

Ejemplo 2

Análisis del programa fuga:

```
# valgrind --tool=memcheck ./fuga
==3273== Memcheck, a memory error detector
==3273== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==3273== Using Valgrind-3.10.0 and LibVEX; rerun with -h for copyright
info
==3273== Command: ./fuga2
==3273==
ÉXITO - Asignación conseguida.
==3273==
==3273== HEAP SUMMARY:
==3273==      in use at exit: 20 bytes in 1 blocks
==3273==    total heap usage: 1 allocs, 0 frees, 20 bytes allocated
==3273==
==3273== LEAK SUMMARY:
==3273==    definitely lost: 20 bytes in 1 blocks
==3273==    indirectly lost: 0 bytes in 0 blocks
==3273==    possibly lost: 0 bytes in 0 blocks
==3273==    still reachable: 0 bytes in 0 blocks
==3273==           suppressed: 0 bytes in 0 blocks
==3273== Rerun with --leak-check=full to see details of leaked memory
==3273==
==3273== For counts of detected and suppressed errors, rerun with: -v
==3273== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Esta vez, descubrimos un problema. Hemos perdido 20 bytes:

```
==3173==    definitely lost: 20 bytes in 1 blocks
```

Ejercicios

Para los dos ejercicios, utilizaremos la máquina virtual Debian. Inicie una sesión en línea de comandos con el usuario root.

1. Matar un programa zombi

1. Escriba el siguiente código en un editor de texto. Asigne al archivo el nombre zombi.c:

```
// zombi.c

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main(void)
{
    pid_t child_pid;

    child_pid = fork ();

    if (child_pid > 0) {
        printf("Proceso padre: Esperar 2 minutos.\n");
        sleep(120);
    }
    else {
        printf("Proceso hijo: Salida inmediata.\n");
        exit(0);
    }
    return(0);
}
```

2. Compile con gcc mediante el siguiente comando:

```
# gcc zombi.c -o zombi
```

3. Ejecute el programa zombi en segundo plano.
4. Dispone de dos minutos. Identifique el PID y el PPID del proceso zombi.
5. Elimine el zombi. Verifique.

2. Identificar una fuga de memoria

1. Escriba el siguiente código en un editor de texto. Asigne al archivo el nombre fuga.c:

```
// fuga1.c

#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    // Asignación de 20 bytes
    char * puntero = malloc(20 * sizeof(char));

    if(puntero == NULL)
    {
        printf("FALLO - No hay asignación.\n");
    }
    else
    {
        printf("ÉXITO - Asignación conseguida.\n");

        //Liberación de los 20 bytes previamente asignados
        free(puntero);

        // borrado del puntero
        puntero = NULL;
    }
}
```

2. Compile con gcc mediante el siguiente comando:

```
# gcc fuga1.c -o fuga1
```

3. Ejecute el programa con valgrind. ¿Existe alguna fuga de memoria?

4. Copie el archivo fuga1.c a fuga2.c.

5. Abra el archivo con un editor de texto.

6. Va a modificar la primera línea. Reemplace fuga1.c por fuga2.c. Luego, comente la línea free (puntero); añadiendo // al inicio de línea.

He aquí el código que debería obtener:

```
// fuga2.c

#include <stdio.h>
#include <stdlib.h>
void main(void)
{
    // Asignación de 20 bytes
    char * puntero = malloc(20 * sizeof(char));

    if(puntero == NULL)
    {
        printf("FALLO - No hay asignación.\n");
    }
    else
    {
        printf("ÉXITO - Asignación conseguida.\n");

        //Liberación de los 20 bytes previamente asignados
        //free(puntero);

        // borrado del puntero
        puntero = NULL;
    }
}
```

7. Compile con gcc fuga2.c. El binario deberá llamarse fuga2.
8. Ejecute el programa con valgrind. ¿Existe alguna fuga de memoria?

Modelos de comunicación

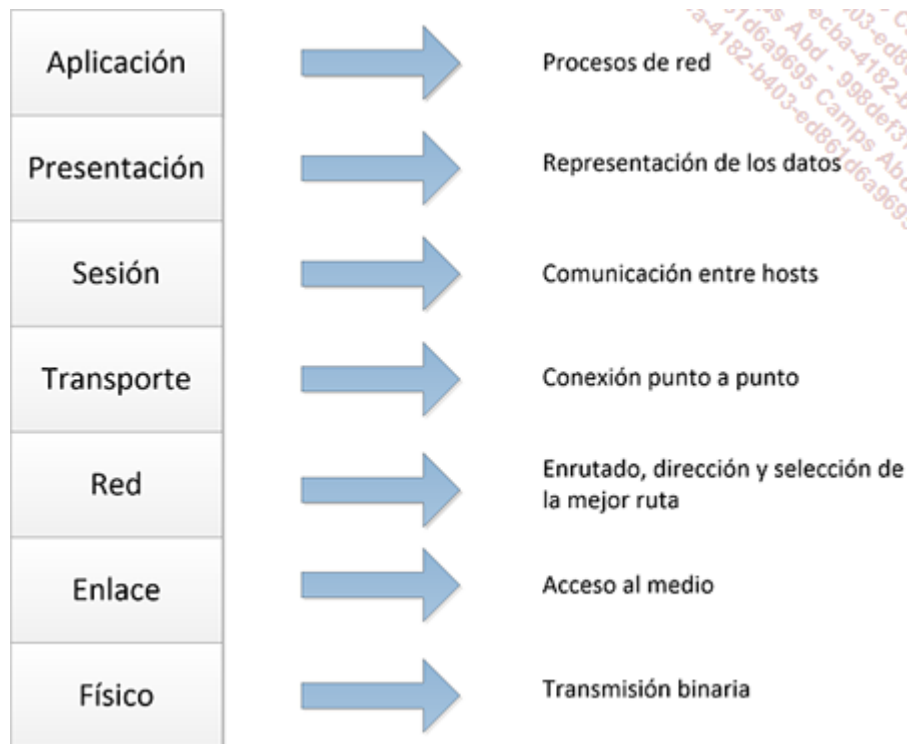
Disponemos de dos modelos de comunicación para ayudarnos a diagnosticar el mal funcionamiento de un equipo Linux conectado a una red.

- El modelo OSI (Open Systems Interconnection).
- El modelo TCP/IP (Transmission Control Protocol/Internet Protocol).

1. Recordatorio del modelo OSI

Con las arquitecturas SNA de IBM, DNA de DEC, Appletalk de Apple por citar algunas, ¿cómo garantizar la interoperabilidad? Ya que cada uno tiene sus propios protocolos de comunicación.

Para evitar la multiplicación de las soluciones de interconexión de arquitecturas heterogéneas, la ISO (International Standards Organization) desarrolló el modelo de comunicación OSI en 1984. Se compone de 7 capas:



El interés de crear un modelo es:

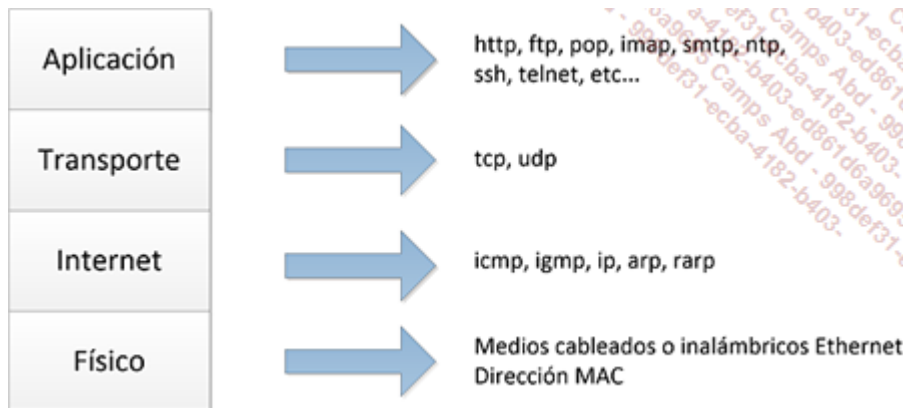
- garantizar la interoperabilidad entre varias tecnologías,

- reducir la complejidad,
- unificar las interfaces,
- permitir una evolución más sencilla,
- facilitar el diseño debido a su modularidad,
- simplificar la enseñanza,
- facilitar la adquisición de conocimientos, etc.

Durante un diagnóstico, podemos basarnos en este modelo, comenzando por la capa física hasta subir a la capa de aplicación.

2. Modelo TCP/IP

El modelo de comunicación TCP/IP, creado antes del modelo OSI, se define según la RFC (Request for Comments) 1122. Describe la pila de protocolos TCP/IP en 4 capas:



La capa física define la interfaz de comunicación del equipo Linux, en particular Ethernet y la dirección MAC, al igual que el controlador del dispositivo.

La capa de Internet define el ámbito de enrutamiento con el protocolo IP. También incluye el protocolo de multicast IGMP, el protocolo de control de mensajes ICMP y el protocolo de resolución de direcciones MAC a direcciones IP, ARP.

La capa de transporte incluye los dos protocolos TCP y UDP que utilizan respectivamente los puertos de comunicación numerados de 0 a 65535.

La capa de aplicación integra las aplicaciones de red tales como SMTP, LDAP, FTP...

3. Modelo OSI vs. modelo TCP/IP

La existencia de dos modelos de comunicación puede parecer extraña para describir la misma cosa. Ambos tienen dos orígenes diferentes y su respectiva historia.

Podemos encontrar coincidencias entre estos dos modelos:

Interfaz física

1. Detección de hardware y controlador

Si el ordenador no tiene ninguna comunicación con la red, podemos empezar a plantearnos estas preguntas:

- ¿Está la tarjeta bien conectada y con alimentación?
- ¿Ha detectado el sistema a la tarjeta de red?
- ¿Existe un controlador en el sistema? En caso afirmativo, ¿está cargado? En caso contrario, ¿lo podemos cargar?
- Si por el contrario, no contamos con el controlador, ¿se encuentra disponible para descargar en Internet? En caso afirmativo, ¿es código fuente para compilar e instalar como módulo del kernel? De lo contrario, habría que desarrollar o simplemente añadir al equipo una interfaz de comunicación compatible con el sistema operativo Linux.

Ejemplo 1

Un equipo, que cuenta con una tarjeta de red Intel, no puede comunicarse. Al utilizar ifconfig desde un terminal, solo se encuentra la interfaz de loopback (siendo lo su nombre lógico).

```
# ifconfig
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      adr inet6: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:12 errors:0 dropped:0 overruns:0 frame:0
      TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 lg file transmission:0
      RX bytes:836 (836.0 B)    TX bytes:836 (836.0 B)
```

Lshw con la opción -c seguida de la palabra clave network enumera el hardware de la clase red:

```
# lshw -c network
*-network UNCLAIMED
      description: Ethernet controller
      product: 82540EM Gigabit Ethernet Controller
      vendor: Intel Corporation
```

```
physical id: 3
bus info: pci@0000:00:03.0
version: 02
width: 32 bits
clock: 66MHz
capabilities: pm pcix cap_list
configuration: latency=64 mingnt=255
resources: memory:f0000000-f001ffff ioport:d010(size=8)
```

La palabra «UNCLAIMED» a la derecha de la palabra Network en la segunda línea significa que la interfaz de red Intel 82540EM Gigabit Ethernet Controller se ha detectado. Sin embargo, no tiene asignado ningún controlador de dispositivo.

No cuenta con un nombre lógico como eth0 ya que este no se indica. También podemos realizar la verificación de esta manera:

```
# ifconfig eth0
eth0: error fetching interface information:
Device not found
```

Como conclusión, el sistema ha detectado el hardware de forma correcta. Solo falta la instalación del controlador compatible con el dispositivo. Si el controlador se encuentra en el sistema, estará situado en el árbol `/lib/modules/$(uname -r)/kernel/drivers/net`. En caso contrario, debemos descargar el controlador desde Internet, compilarlo como módulo del kernel y luego instalarlo.

Consulte el capítulo Módulos.

Ejemplo 2

Tomemos el mismo equipo que en el ejemplo 1, siempre con una tarjeta de red Intel. El resultado del comando `ifconfig` indica que las interfaces `eth0` y `loopback` están presentes:

```
# ifconfig
eth0  Link encap:Ethernet  HWaddr 08:00:27:1c:fd:f5
      inet addr:192.168.1.14  Bcast:192.168.1.255
      Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe1c:fd5/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:236699 errors:0 dropped:0 overruns:0 frame:0
      TX packets:101578 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
```

```
RX bytes:356945522 (340.4 MiB) TX bytes:6928362 (6.6 MiB)
```

```
lo    Link encap:local loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:43 errors:0 dropped:0 overruns:0 frame:0
TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:4803 (4.6 KiB) TX bytes:4803 (4.6 KiB)
```

Con lshw, la lista del hardware de la clase red se presenta así:

```
# lshw -c network
```

```
*-network
```

```
description: Ethernet interface
product: 82540EM Gigabit Ethernet Controller
vendor: Intel Corporation
physical id: 3
bus info: pci@0000:00:03.0
logical name: eth0
version: 02
serial: 08:00:27:dbb:f:ad
size: 1GB/s
capacity: 1GB/s
width: 32 bits
clock: 66MHz
```

```
capabilities: pm pcix bus_master cap_list ethernet physical
tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
configuration: autonegotiation=on broadcast=yes driver=e1000
driverversion=7.3.21-k5-NAPI duplex=full firmware=N/A ip=10.0.2.15
latency=64 link=yemingnt=255 multicast=yes port=twisted pair

speed=1GB/s
```

```
resources: irq:19 memory:f0000000-f001ffff
ioport:d010(size=8)
```

La interfaz de red Intel 82540EM Gigabit Ethernet Controller no se encuentra «UNCLAIMED» por un lado mientras que cuenta con el nombre lógico eth0.

El controlador de dispositivo e1000 está instalado de forma correcta ya que podemos leer driver=e1000. Se almacena en la carpeta /lib/modules/`uname -r`/kernel/drivers/net/ethernet/Intel/e1000:

```
ls -l /lib/modules/`uname -r`/kernel/drivers/net/ethernet/intel
/e1000
total 140
-rw-r--r-- 1 root root 139328 jun  9 20:44 e1000.ko
```

Utilizando lsmod y grep podemos saber si el controlador está cargado:

```
# lsmod |grep e1000
e1000                86156  0
```

2. Direccionamiento físico

La dirección MAC (Media Access Control) identifica de forma única la interfaz de comunicación en 48 bits, o sea 6 bytes. Está representada en hexadecimal de la siguiente forma: 08:00:27:1A:1A:48.

a. Identificar la dirección MAC

Varios comandos permiten obtener la dirección MAC de una tarjeta de red. Hasta ahora hemos hablado de lshw o ifconfig:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1c:fd:f5
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1c:fd5/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0

          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000
```

```

RX bytes:938 (938.0 B)  TX bytes:10461 (10.2 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:b6:27:6a
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3563 (3.4 KiB)  TX bytes:3563 (3.4 KiB)

```

Para obtener la dirección de una tarjeta de red, incluimos su nombre lógico con ifconfig.

Sintaxis

```
ifconfig <interface>
```

Este comando no tiene opciones.

Ejemplo

Mostrar la configuración de la interfaz eth1:

```

# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:b6:27:6a
          inet adr:192.168.56.101  Bcast:192.168.56.255
          Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1c:fd5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

```

```
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

También podemos usar el comando `ip addr`:

Ejemplo

Mostrar la configuración de todas las interfaces:

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:3a:17:d8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe3a:17d8/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:8b:0d:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8b:d9d/64 scope link
        valid_lft forever preferred_lft forever
```

El comando `ip addr` dispone de la opción `show` para ver los parámetros de una interfaz en particular:

Ejemplo

Mostrar la configuración de la interfaz `eth1`:

```
# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8b:0d:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8b:d9d/64 scope link
        valid_lft forever preferred_lft forever
```

b. Caché ARP

La caché ARP es una tabla que contiene las direcciones MAC asociadas a las direcciones IP. Así, obtenemos la correspondencia entre la dirección hardware y la dirección software o lógica.

Cuando un equipo se comunica con otro equipo de la misma subred (subnet), debe resolver la dirección lógica (dirección IP) por su dirección hardware (dirección MAC). Busca en su caché ARP. Si la resolución no se realiza, se envía una solicitud ARP (Address Resolution Protocol) a todos los hosts de la subred: «¿Quién tiene la dirección IP w.x.y.z?». El host en cuestión responde a la solicitud. Los dos hosts actualizan su caché ARP respectiva.

El comando arp gestiona la caché ARP. Toma su información del archivo /proc/net/arp.

Sintaxis

```
arp <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-a <host>	--display <host>	Muestra el host solicitado.
-v	--verbose	Modo extendido.
-n	--numeric	Muestra las direcciones en lugar de los nombres de host.

-s <host> <dir>	--set <host> <dir>	Añade manualmente una entrada en la tabla ARP.
-d <host>	--delete <host>	Elimina una entrada de la tabla ARP.

Ejemplo

Añadir manualmente la dirección IP 10.0.2.99 de un host con la dirección MAC asociada:

```
# arp -s 10.0.2.99 52:54:00:12:35:99
```

Verificar la adición:

```
# arp
Address      HWtype  HWaddress      Flags Mask    Iface
10.0.2.3     ether   52:54:00:12:35:03  C             eth0
10.0.2.99    ether   52:54:00:12:35:99  CM            eth0
```

La primera dirección IP 10.0.2.3 es una entrada dinámica. Nuestra entrada manual se perderá tras el próximo arranque del equipo.

Para eliminar una entrada:

```
# arp -d 10.0.2.99
```

La entrada aparecerá así:

```
# arp
Address      HWtype  HWaddress      Flags Mask    Iface
10.0.2.3     ether   52:54:00:12:35:03  C             eth0
10.0.2.99    ether   (incomplete)          eth0
```

La entrada desaparecerá después en un espacio de tiempo breve.

c. ARPing

El comando arping prueba la conectividad de un equipo enviando comandos de tipo ARP. Es similar al comando ping que utiliza una petición de tipo ICMP (Internet Control Message Protocol).

Esta herramienta de red no está instalada por defecto en Debian y Ubuntu, pero se encuentra en los repositorios. Con CentOS está instalada por defecto.

Instalación en Debian y Ubuntu

La instalación se realiza así:

```
# apt-get -y install arping
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libnet1
Se instalarán los siguientes paquetes NUEVOS:
  arping libnet1
0 actualizados, 2 nuevos se instalarán, 1 para eliminar y 0 no
actualizados.
Se necesita descargar 86,5 kB de archivos.
Se utilizarán 229 kB de espacio de disco adicional después de
esta operación.
```

Conteste s para continuar:

```
Des:1 http://ftp.es.debian.org/debian/ jessie/main libnet1
amd64 1.1.6+dfsg-3 [60,4 kB]
Des:2 http://ftp.es.debian.org/debian/ jessie/main arping
amd64 2.14-1 [26,1 kB]
Descargados 86,5 kB en 0s (597 kB/s)
(Leyendo la base de datos ... 248217 ficheros o directorios
instalados actualmente.)
Desinstalando iputils-arping (3:20121221-5+b2) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Seleccionando el paquete libnet1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 248211 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../libnet1_1.1.6+dfsg-3_amd64.deb ...

Desempaquetando libnet1:amd64 (1.1.6+dfsg-3) ...
Seleccionando el paquete arping previamente no seleccionado.
Preparando para desempaquetar .../arping_2.14-1_amd64.deb ...
```

```
Desempaquetando arping (2.14-1) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Configurando libnet1:amd64 (1.1.6+dfsg-3) ...
Configurando arping (2.14-1) ...
```

Sintaxis

```
arping <opciones> <host_objetivo>
```

Opciones necesarias:

Opción	Descripción
-I <interface>	Especifica la interfaz que va a enviar las peticiones.
-c <valor>	Define el número de peticiones a enviar.

Ejemplo

```
# arping -I eth1 -c 3 192.168.56.1
ARPING 192.168.56.1
60 bytes from 0a:00:27:00:00:00 (192.168.56.1): index=0
time=106.152 usec
60 bytes from 0a:00:27:00:00:00 (192.168.56.1): index=1
time=112.035 usec
60 bytes from 0a:00:27:00:00:00 (192.168.56.1): index=2
time=99.451 usec
```

Interfaz lógica

Si bien los controladores de dispositivo de bloques y caracteres se montan como archivos en la carpeta /dev, esto no ocurre para las interfaces de red. Se accede a estos últimos a través de un nombre lógico como eth0, eth1. Eth haciendo referencia a Ethernet y el valor numérico 0 para la primera tarjeta, 1 para la segunda tarjeta y así sucesivamente... Según la distribución, una interfaz inalámbrica puede tener un nombre lógico eth0 o wlan0. Wlan hace referencia a wireless LAN. Los nombres se obtienen del administrador de dispositivos udev.

1. CentOS 7: cambiar el nombre de la interfaz

A partir de la versión 7, CentOS utiliza otros nombres lógicos que hemos comentado en el capítulo Arquitectura del sistema GNU/Linux. Podemos cambiar el modelo de nombres y volver a los nombres lógicos ethX.

Ejemplo

Abra el archivo /etc/default/grub con un editor de texto.

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="vconsole.keymap=es rd.lvm.lv=sysVG/swap
vconsole.font=latarcyrheb-sun16 crashkernel=auto rd.lvm.lv=sysVG/
root rhgb quiet net.ifnames=0"
GRUB_DISABLE_RECOVERY="true"
```

Busque la línea que empieza por GRUB_CMDLINE_LINUX y agregue al final de esta línea net.ifnames=0. Luego, guarde el archivo.

Regenere la configuración de GRUB escribiendo:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-229.14.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-
229.14.1.el7.x86_64.img
```

```
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-
85d58cfe9e95420caf192d072326775e
Found initrd image: /boot/initramfs-0-rescue-
85d58cfe9e95420caf192d072326775e.img
done
```

En el directorio `/etc/sysconfig/network-scripts/`, hay que renombrar el archivo de configuración de cada interfaz. Por ejemplo, si tenemos dos interfaces `enp0s3` y `enp0s8`, se convertirán respectivamente en `eth0` y `eth1`.

```
# mv ifcfg-enp0s3 ifcfg-eth0
# mv ifcfg-enp0s8 ifcfg-eth1
```

Modifique con `sed` en cada archivo el valor del campo `name`:

```
# sed -i 's/enp0s3/eth0/g' ifcfg-eth0
# sed -i 's/enp0s8/eth1/g' ifcfg-eth1
```

El comando `udevadm info` muestra información de un dispositivo a partir de la base de datos `udev`.

Sintaxis

```
udevadm info <opciones>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-a	--attribute-walk	Muestra las propiedades de sysfs correspondientes al dispositivo especificado.
-p	--path	Especifica el dispositivo a mostrar en sysfs.

Ejemplo

```
# udevadm info -a -p /sys/class/net/eth0/
```

Udevadm info starts with the device specified by the devpath and then walks up the chain of parent devices. It prints for every device found, all possible attributes in the udev rules key format. A rule to match, can be composed by the attributes of the device and the attributes from one single parent device.

```
looking at device '/devices/pci0000:00/0000:00:03.0/net/eth0':
```

```
  KERNEL=="eth0"
  SUBSYSTEM=="net"
  DRIVER=="
  ATTR{addr_assign_type}=="0"
  ATTR{addr_len}=="6"
  ATTR{dev_id}=="0x0"
  ATTR{ifalias}=="
  ATTR{iflink}=="2"
  ATTR{ifindex}=="2"
  ATTR{type}=="1"
  ATTR{link_mode}=="0"
  ATTR{address}=="08:00:27:1a:1a:48"
  ATTR{broadcast}=="ff:ff:ff:ff:ff:ff"
  ATTR{carrier}=="1"
  ATTR{speed}=="1000"
  ATTR{duplex}=="full"
  ATTR{dormant}=="0"
  ATTR{operstate}=="up"
  ATTR{mtu}=="1500"
  ATTR{flags}=="0x1003"
  ATTR{tx_queue_len}=="1000"
  ATTR{netdev_group}=="0"
```

```
looking at parent device '/devices/pci0000:00/0000:00:03.0':
```

```

KERNELS=="0000:00:03.0"
SUBSYSTEMS=="pci"
DRIVERS=="e1000"
ATTRS{vendor}=="0x8086"
ATTRS{device}=="0x100e"
ATTRS{subsystem_vendor}=="0x8086"
ATTRS{subsystem_device}=="0x001e"
ATTRS{class}=="0x020000"
ATTRS{irq}=="19"

ATTRS{local_cpus}=="00000000,00000000,00000000,00000000,00000000,
00000000,000000,00000000,00000000,00000000,00000000,00000000,00000000,
00000000,00000000,00000000,00000003"
ATTRS{local_cpulist}=="0-1"
ATTRS{numa_node}=="-1"
ATTRS{dma_mask_bits}=="32"
ATTRS{consistent_dma_mask_bits}=="32"
ATTRS{enable}=="1"
ATTRS{broken_parity_status}=="0"
ATTRS{msi_bus}=="

looking at parent device '/devices/pci0000:00':
KERNELS=="pci0000:00"
SUBSYSTEMS=="
DRIVERS=="

```

2. Activar/desactivar una interfaz

El comando `ifconfig` también permite activar o desactivar una interfaz.

Sintaxis

```
ifconfig <interfaz> <opciones>
```

Opciones necesarias:

<https://dogramcode.com/libros-sistemas>

Opción	Descripción
up	Activa la interfaz mencionada.
down	Desactiva la interfaz mencionada.

Ejemplo

Desactivar la interfaz eth0:

```
#ifconfig eth0 down
```

Activar la interfaz eth0:

```
# ifconfig eth0 up
```

ifconfig no negocia un contrato DHCP ni renegocia el contrato en curso. En este caso, debemos usar los comandos ifup e ifdown.

Sintaxis

```
ifup <interface>
ifdown <interface>
ifup -a
```

Opciones útiles:

Opción		Descripción
corta	larga	
-a	--all	Activa todas las interfaces definidas en el archivo /etc/network/interfaces.

Ejemplo

Activar la interfaz eth0:

```
# ifup eth0

Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.

All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
```

```
Listening on LPF/eth0/08:00:27:1a:1a:48
Sending on   LPF/eth0/08:00:27:1a:1a:48
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPOFFER from 10.0.2.2
DHCPACK from 10.0.2.2
bound to 10.0.2.15 -- renewal in 35932 seconds.
```

La trama DHCPDISCOVER en el puerto UDP 67 realiza un broadcast utilizando la dirección 255.255.255.255 para encontrar un servidor DHCP disponible, el cual responde con la trama DHCPREQUEST. La máquina Linux acepta la oferta mediante la trama DHCPOFFER. El intercambio concluye con el servidor DHCP enviando la trama DHCPACK.

Desactivar la interfaz eth0:

```
# ifdown eth0

Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/08:00:27:1a:1a:48
Sending on   LPF/eth0/08:00:27:1a:1a:48
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 10.0.2.2 port 67
```

La trama DHCPRELEASE libera el contrato.

3. Configurar una interfaz

La configuración de las interfaces de red es diferente. Tenemos por un lado Debian y Ubuntu y por el otro CentOS.

a. Configuración de interfaces en Debian y Ubuntu

La configuración se encuentra en un único archivo llamado interfaces situado en la carpeta /etc/network:


```
# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

# The 2nd network interface
auto eth1
iface eth1 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    gateway 192.168.1.254
```

El siguiente cuadro explica las entradas de /etc/network/interfaces:

Entrada	Descripción
auto <interfaz>	Arranca la interfaz durante el boot.
iface <nombre_config> inet loopback	La configuración de la interfaz loopback (127.0.0.1).
iface <nombre_config> inet dhcp	La configuración IP es dinámica (cliente DHCP).
iface <nom_config> inet static	La configuración IP es estática. Las opciones address, netmask y gateway son necesarias para definir los parámetros.
pre-up	Ejecuta un comando antes de activar la interfaz.

post-up	Ejecuta un comando después de activar la interfaz.

b. Configuración de las interfaces en CentOS

Existe un archivo de configuración por cada interfaz en la carpeta `/etc/sysconfig/network-scripts`. El archivo se llamará de la siguiente manera `ifcfg-<interfaz>`:

```
# ls -l /etc/sysconfig/network-scripts/ifcfg*
/etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-lo
```

He aquí el contenido del archivo de la interfaz de loopback:

```
# cat /etc/sysconfig/network-scripts/ifcfg-lo
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
# If you're having problems with gated making 127.0.0.0/8 a martian,
# you can change this to something else (255.255.255.255, for example)
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

Del mismo modo, para la interfaz `eth1`:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
HWADDR=08:00:27:61:1B:3F
TYPE=Ethernet
UUID=ef71420f-0c55-4409-ba89-9a347e11e811

ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

El siguiente cuadro explica las entradas de los archivos de configuración de red:

Entrada	Descripción
NAME	Nombre lógico del dispositivo.
IPADDR	Dirección IP.
NETMASK	Máscara de subred.
NETWORK	Dirección de red.
BROADCAST	Dirección de broadcast.
ONBOOT	Activado si yes; desactivado si no.
HWADDR	Dirección hardware (dirección MAC).
GATEWAY	Dirección de la puerta de enlace por defecto.
DNS1 DNS2	Dirección IP del servidor DNS primario. Dirección IP del servidor DNS secundario.
PEERDNS	Si yes, modifica el archivo /etc/resolv.conf si las directivas DNS están definidas. El valor por defecto es yes si usamos un cliente DHCP. Si no, no modifica el archivo /etc/resolv.conf.
NM_CONTROLLED	Si yes, el dispositivo se gestiona mediante el demonio Network Manager. Si no, el dispositivo se gestiona de forma manual mediante su archivo de configuración.
BOOTPROTO	Indica el protocolo de arranque: none : ningún protocolo. dhcp : usar el protocolo DHCP.
USERCTL	Si yes, todos los usuarios pueden gestionar la interfaz de red.

	Si no, solo el usuario root puede gestionar la interfaz.
ETHTOOL_OPTS	Define las opciones Ethernet. Por ejemplo: ETHTOOL_OPTS="autoneg off speed 1000 duplex full" El valor ON u OFF de autoneg activa o desactiva la negociación automática. speed especifica la velocidad 10, 100 o 1000 en Mb/s. duplex permite el valor full o half.

c. Herramienta ethtool

La herramienta ethtool gestiona la configuración de las interfaces de red. Esta herramienta no está instalada por defecto en Debian y Ubuntu, a diferencia de CentOS.

Instalación en Debian y Ubuntu

La instalación del paquete ethtool:

```
# apt-get install ethtool
```

Sintaxis

```
ethtool <interface>
ethtool -s <interfaz> <opciones>
ethtool -S <interfaz>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-s	--	Establece los parámetros tales como dúplex y la velocidad de la interfaz mencionada.
-S	-- statistics	Devuelve estadísticas de la interfaz especificada.

Ejemplos:

Mostrar la configuración de la interfaz eth0:

```
# ethtool eth0
```

```
Settings for eth0:
```

```
Supported ports: [ TP ]
```

```
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
```

```
Supported pause frame use: No
```

```
Supports auto-negotiation: Yes
```

```
Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
```

```
Advertised pause frame use: No
```

```
Advertised auto-negotiation: Yes
```

```
Speed: 1000Mb/s
```

```
Duplex: Full
```

```
Port: Twisted Pair
```

```
PHYAD: 0
```

```
Transceiver: internal
```

```
Auto-negotiation: on
```

```
MDI-X: Unknown
```

```
Supports Wake-on: umbg
```

```
Wake-on: d
Current message level: 0x00000007 (7)
                drv probe link
Link detected: yes
```

Especificar la velocidad de la interfaz a 1000 Mb/s en full duplex:

```
# ethtool -s eth0 autoneg off speed 1000 duplex full
```

Para que los parámetros sean permanentes en Debian y Ubuntu, debemos añadir la siguiente línea en /etc/network/interfaces:

```
pre-up /usr/sbin/ethtool -s $IFACE autoneg off 1000 duplex full
```

En cuanto a CentOS, añadimos la siguiente línea en /etc/sysconfig/network-scripts/ifcfg-eth0:

```
ETHTOOL_OPTS="speed 1000 duplex full autoneg off"
```

d. Interfaz virtual

Podemos crear una interfaz virtual para asociar una segunda dirección IP a una tarjeta de red. Basta con añadir al nombre lógico un :0 para la primera interfaz virtual, un :1 para la segunda y así sucesivamente... Por ejemplo: eth0:0.

Ejemplo con Debian y Ubuntu

Añadir en el archivo /etc/network/interfaces:

```
iface eth0:0 inet static
    address 192.168.1.200
    netmask 255.255.255.0
    gateway 192.168.1.254
```

Ejemplo con CentOS

Vaya a la carpeta /etc/sysconfig/network-scripts y copie el archivo ifcfg-eth0 en ifcfg-eth0:0:

```
# cp ifcfg-eth0 ifcfg-eth0:0
```

Verifique:

```
# ls -l ifcfg-eth*
ifcfg-eth0
ifcfg-eth0:0
```

El archivo ifcfg-eth0 contiene lo siguiente:

```
# cat ifcfg-eth0
DEVICE=eth0
HWADDR=08:00:27:61:1B:3F
TYPE=Ethernet
UUID=ef71420f-0c55-4409-ba89-9a347e11e811
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

No queda más que modificar los parámetros de la interfaz virtual en el archivo ifcfg-eth0:0 con un editor de texto:

```
# cat ifcfg-eth0:0
DEVICE=eth0:0
TYPE=Ethernet
BOOTPROTO=static
IPADDR=10.0.0.150
NETMASK=255.255.255.0
GATEWAY=10.0.0.1
BROADCAST=10.0.0.255
NETWORK=10.0.0.0
USERCTL=no
ONBOOT=yes
```

A continuación, reiniciamos el servicio de red:

```
# systemctl restart network
```

Si utilizamos el comando service de versiones anteriores de CentOS, observaremos que este llama a systemctl:

```
# service network restart
Restarting network (via systemctl): [ OK ]
```

Verificamos:

```
# ifconfig eth0:0
eth0:0 Link encap:Ethernet HWaddr 08:00:27:61:1B:3F
```

```
inet adr:10.0.0.150 Bcast:10.0.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

4. Direccionamiento IPv4

Para ayudarnos a calcular una dirección IP, podemos usar ipcalc. Esta herramienta no se encuentra por defecto en ninguna de las tres distribuciones y debe ser instalada.

Debian y Ubuntu

```
# apt-get install ipcalc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  ipcalc
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 26,3 kB de archivos.
Se utilizarán 110 kB de espacio de disco adicional después de esta
operación.
Des:1 http://ftp.es.debian.org/debian/ jessie/main
ipcalc all 0.41-4 [26,3 kB]
Descargados 26,3 kB en 0s (286 kB/s)
Seleccionando el paquete ipcalc previamente no seleccionado.
(Leyendo la base de datos ... 248153 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../archives/ipcalc_0.41-
4_all.deb ...
Desempaquetando ipcalc (0.41-4) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Configurando ipcalc (0.41-4) ...
```

CentOS

En cuanto a CentOS, el comando ipcalc no es igual. Podemos descargar otra versión:

<https://dogramcode.com/libros-sistemas>


```
# wget http://jodies.de/ipcalc-archive/ipcalc-0.41.tar.gz
--2016-05-04 14:46:20-- http://jodies.de/ipcalc-archive/
ipcalc-0.41.tar.gz
Resolviendo jodies.de (jodies.de)... 193.175.80.168
Conectando con jodies.de (jodies.de)[193.175.80.168]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 21599 (21K) [application/x-gzip]
Grabando a: "ipcalc-0.41.tar.gz"

100%[=====>] 21.599      --.-K/s  en 0,1s

2016-05-04 14:46:20 (187 KB/s) - "ipcalc-0.41.tar.gz" guardado
[21599/21599]
```

Como es un archivo tar, puede descomprimirse en la carpeta /opt:

```
# tar xzvf ipcalc-0.41.tar.gz
ipcalc-0.41/
ipcalc-0.41/ipcalc
ipcalc-0.41/changelog
ipcalc-0.41/ipcalc.cgi
ipcalc-0.41/ipcalculator.png
ipcalc-0.41/license
ipcalc-0.41/contributors
ipcalc-0.41/ipcalc.gif
```

Para probar el comando, entramos en la carpeta /opt/ipcalc-0.41:

```
# cd ipcalc-0.41/
```

Luego invocamos el comando con la opción -v para ver la versión:

```
# ./ipcalc -v
0.41
```

Sintaxis

```
ipcalc <dirección_IP> <máscara>
```

Este comando no tiene opciones.

Ejemplo

Tenemos una dirección IP 172.16.33.1/19. Queremos saber en qué subred nos encontramos:

```
# ipcalc 172.16.33.1/19
Address: 172.16.33.1      10101100.00010000.001 00001.00000001
Netmask: 255.255.224.0 = 19 11111111.11111111.111 00000.00000000
Wildcard: 0.0.31.255      00000000.00000000.000 11111.11111111
=>
Network: 172.16.32.0/19   10101100.00010000.001 00000.00000000
HostMin: 172.16.32.1     10101100.00010000.001 00000.00000001
HostMax: 172.16.63.254   10101100.00010000.001 11111.11111110
Broadcast: 172.16.63.255 10101100.00010000.001 11111.11111111
Hosts/Net: 8190          Class B, Private Internet
```

a. Dirección IPv4 estática

En este capítulo hemos indicado que la dirección IP se almacena en un archivo de configuración y que ifconfig permite consultar la dirección MAC y la dirección IP.

ifconfig también permite asignar sobre la marcha una dirección IP.

Sintaxis

```
ifconfig <interfaz> <dirección> netmask <máscara>
ifconfig <interfaz> <dirección/máscara>
```

Este comando no tiene opciones.

Ejemplo

Deseamos asignar puntualmente la dirección IP 172.18.65.3/19 a la interfaz eth0. Con ipcalc veremos las características de esta dirección:

```
# ipcalc 172.18.65.3/19
Address: 172.18.65.3      10101100.00010010.010 00001.00000011
Netmask: 255.255.224.0 = 19 11111111.11111111.111 00000.00000000
Wildcard: 0.0.31.255      00000000.00000000.000 11111.11111111
=>
Network: 172.18.64.0/19   10101100.00010010.010 00000.00000000
HostMin: 172.18.64.1     10101100.00010010.010 00000.00000001
```

```
HostMax: 172.18.95.254      10101100.00010010.010 11111.11111110
Broadcast: 172.18.95.255    10101100.00010010.010 11111.11111111
Hosts/Net: 8190              Class B, Private Internet
```

Ahora, asignamos la dirección a la interfaz eth0:

```
# ifconfig eth0 172.18.65.3/19
```

Comprobamos el resultado:

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:1a:48
          inet adr:172.18.65.3  Bcast:172.18.95.255
Mask:255.255.224.0
          adr inet6: fe80::a00:27ff:fe1a:1a48/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:857 errors:0 dropped:0 overruns:0 frame:0
          TX packets:885 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:749507 (731.9 KiB)  TX bytes:90277 (88.1 KiB)
```

b. Direccionamiento IPv4 dinámico

Durante la ejecución de ifup y de ifdown, se negocia un contrato DHCP además de la activación de la interfaz. Si solo queremos la negociación de un contrato, contamos con el comando dhclient.

Sintaxis

```
dhclient <interfaz> <opción>
```

Opción útil:

Opción corta	Descripción
-v	Modo extendido.

Ejemplo

Desea que la interfaz eth0 contacte de nuevo con un servidor DHCP:

```
# ifdown eth0
```

```
Internet Systems Consortium DHCP Client 4.2.2
```

```
Copyright 2004-2011 Internet Systems Consortium.
```

```
All rights reserved.
```

```
For info, please visit https://www.isc.org/software/dhcp/
```

```
Listening on LPF/eth0/08:00:27:1a:1a:48
```

```
Sending on   LPF/eth0/08:00:27:1a:1a:48
```

```
Sending on   Socket/fallback
```

```
DHCPREQUEST on eth0 to 255.255.255.255 port 67
```

```
DHCPACK from 10.0.2.2
```

```
RTNETLINK answers: File exists
```

```
bound to 10.0.2.15 -- renewal in 40872 seconds.
```

c. Eliminar la dirección IPv4

Para eliminar la dirección IP asignada a una interfaz, empleamos `ip addr flush`.

Sintaxis

```
ip addr flush <interfaz>
```

Este comando no tiene opciones.

Ejemplo

Eliminar la dirección de la interfaz `eth0`:

```
# ip addr flush eth0
```

Comprobamos con `ifconfig`:

```
# ifconfig eth0
```

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:1a:48
```

```
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
          RX packets:863 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:922 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 lg file transmission:1000
```

RX bytes:751987 (734.3 KiB) TX bytes:96125 (93.8 KiB)

5. Gateway por defecto

El gateway por defecto es simplemente la dirección que permite al equipo comunicarse con otras redes. Para conocer las rutas de un equipo Linux usamos los comandos `netstat`, `route` o `ip route`.

Sintaxis

```
netstat -r
```

Este comando no tiene opciones.

Sintaxis

```
route -n
```

Este comando no tiene opciones.

Sintaxis

```
ip route
```

Este comando no tiene opciones.

Ejemplos:

Con `netstat`:

```
# netstat -r
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	10.0.3.2	0.0.0.0	UG	0	0	0	eth1
default	Livebox	0.0.0.0	UG	0	0	0	eth0
10.0.3.0	*	255.255.255.0	U	0	0	0	eth1
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0

Con `route`:

```
# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.0.3.2	0.0.0.0	UG	0	0	0	eth1
0.0.0.0	192.168.1.1	0.0.0.0	UG	1024	0	0	eth0

10.0.3.0	0.0.0.0	255.255.255.0	U	0	0	0 eth1
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0 eth0

Para acceder a la red 10.0.3.0/24 a través de la interfaz eth1, el equipo no utilizará la puerta de enlace.

Para acceder a la red 192.168.1.0/24 a través de la interfaz eth0, el equipo tampoco utilizará la puerta de enlace.

Para las demás redes, la pasarela utilizada por defecto será 10.0.3.2.

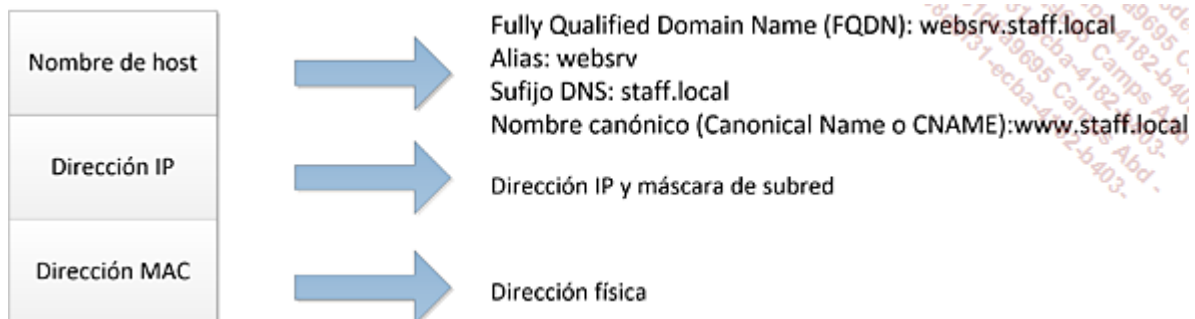
Con ip route:

```
# ip route
default via 10.0.3.2 dev eth1
default via 192.168.1.1 dev eth0 proto static metric 1024
10.0.3.0/24 dev eth1 proto kernel scope link src 10.0.3.15
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.14
```

6. Resolución de nombres de hosts

Una dirección MAC está vinculada a una dirección IP. La función del protocolo ARP es resolver una dirección lógica a una dirección física. Este funcionamiento es intrínseco a TCP/IP.

Una dirección IP está vinculada a un número de hosts. La resolución de nombres se efectuará a través de un archivo /etc/hosts o por un cliente DNS que interroga a través de la red a un servidor DNS.



El FQDN es el nombre completo del equipo mientras que el alias es su nombre relativo. El sufijo DNS representa el espacio de nombres en el cual encontramos al equipo. La concatenación del alias y del sufijo DNS nos devuelve el FQDN.

El nombre canónico señala un alias y proporciona un nombre adicional. De esta forma, un equipo Linux puede tener diferentes nombres correspondientes a los servicios de red.

a. Probar la resolución de nombres

Las herramientas dig y nslookup comprueban la resolución de nombres en direcciones IP.

Sintaxis

```
dig <nombre>
```

Este comando no tiene opciones.

Sintaxis

```
nslookup <nombre>
```

Este comando no tiene opciones.

Ejemplos:

Resolución del nombre www.google.com con dig:

```
# dig www.google.com

; <>> DiG 9.9.5-9+deb8u6-Debian <>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42861
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000

;; QUESTION SECTION:
;www.google.com.          IN      A

;; ANSWER SECTION:
www.google.com.          83      IN      A      216.58.214.100
;; Query time: 5 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed May 04 18:21:52 CEST 2016
;; MSG SIZE rcvd: 59
```

Resolución del nombre www.google.com con nslookup:

```
# nslookup www.google.com
```

```
Server:      8.8.8.8
```

```
Address:     8.8.8.8#53
```

```
Non-authoritative answer:
```

```
Name:   www.google.com
```

```
Address: 173.194.66.105
```

```
Name:   www.google.com
```

```
Address: 173.194.66.106
```

```
Name:   www.google.com
```

```
Address: 173.194.66.147
```

```
Name:   www.google.com
```

```
Address: 173.194.66.99
```

```
Name:   www.google.com
```

```
Address: 173.194.66.103
```

```
Name:   www.google.com
```

```
Address: 173.194.66.104
```

b. Archivos de configuración

El archivo `/etc/resolv.conf` contiene las direcciones de los servidores DNS que el cliente DNS va a utilizar:

```
# cat /etc/resolv.conf
```

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

El archivo `/etc/nsswitch.conf` determina el orden de resolución:

```
# grep hosts /etc/nsswitch.conf
```

```
hosts:          files mdns4_minimal [NOTFOUND=return] dns mdns4
```


Los argumentos se tratan de izquierda a derecha para la directiva hosts::

Argumento	Descripción
files	Lee los nombres con direcciones estáticas en el archivo /etc/hosts.
mdns4_minimal	Resuelve los nombres utilizando multicast DNS.
[NOTFOUND=return]	Si el proceso mdns4_mínimo devuelve una respuesta "no encontrado" (NOTFOUND) entonces el sistema no debe seguir buscando una respuesta.
dns	Efectúa una consulta DNS unicast.
mdns4	Efectúa una consulta DNS multicast.

Herramientas de análisis

1. Problemas de conexión

Si su equipo Linux no consigue comunicarse por la red, contamos con el comando ping. Podemos escuchar a los intervinientes intercambiar la palabra ping, por ejemplo: «¿Conseguimos hacer ping al servidor?».

Sintaxis

```
ping <opción> <dirección_IP>
```

Opción útil:

Opción	Descripción
-c	Especifica el número de peticiones ICMP ECHO REQUEST y ECHO REPLY.

El diagnóstico puede realizarse en varios niveles para verificar:

- la configuración TCP/IP,
- el enlace del equipo a la red,
- la comunicación local,
- la comunicación remota.

a. Verificar la configuración TCP/IP

Si el cable de red, la tarjeta de red y su controlador no parecen presentar problemas, utilizamos ping con la dirección de bucle (127.0.0.1) para comprobar que el protocolo TCP/IP esté configurado de forma correcta.

Ejemplo

```
# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.069 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.048 ms
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_req=4 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_req=5 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_req=6 ttl=64 time=0.041 ms
^C
--- 127.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4997ms
rtt min/avg/max/mdev = 0.036/0.045/0.069/0.014 ms
```

El comando ping no se detiene por defecto, hay que interrumpirlo con la combinación de teclas [Ctrl] + C.

b. Verificar la conexión a la red

Hacemos ping a la dirección IP del equipo local para comprobar si el acceso a la red es correcto.

Ejemplo

Identificamos la dirección IP de la interfaz de red con ifconfig:

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:1a:48
          inet adr:10.0.2.15  Bcast:10.0.2.255
Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fe1a:1a48/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:6 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:1435 (1.4 KiB) TX bytes:10741 (10.4 KiB)
```

A continuación hacemos ping:

```
# ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_req=1 ttl=64 time=0.245 ms
64 bytes from 10.0.2.15: icmp_req=2 ttl=64 time=0.107 ms
64 bytes from 10.0.2.15: icmp_req=3 ttl=64 time=0.042 ms
64 bytes from 10.0.2.15: icmp_req=4 ttl=64 time=0.092 ms
^C
--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.042/0.121/0.245/0.076 ms
```

c. Verificar la comunicación local

Hacemos ping a la dirección IP del gateway por defecto con el fin de verificar la comunicación con un host local en la LAN y al mismo tiempo el acceso a la pasarela.

Ejemplo

Identificamos la dirección de la puerta de enlace por defecto:

```
# route -n
Kernel IP routing table

```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.0.3.2	0.0.0.0	UG	0	0	0	eth1
0.0.0.0	192.168.1.1	0.0.0.0	UG	1024	0	0	eth0
10.0.3.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

Hacemos ping al gateway por defecto:

```
# ping 10.0.3.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
```

```
64 bytes from 10.0.2.2: icmp_req=1 ttl=63 time=0.771 ms
64 bytes from 10.0.2.2: icmp_req=2 ttl=63 time=0.467 ms
64 bytes from 10.0.2.2: icmp_req=3 ttl=63 time=0.723 ms
64 bytes from 10.0.2.2: icmp_req=4 ttl=63 time=0.466 ms
64 bytes from 10.0.2.2: icmp_req=5 ttl=63 time=0.358 ms
64 bytes from 10.0.2.2: icmp_req=6 ttl=63 time=0.418 ms
64 bytes from 10.0.2.2: icmp_req=7 ttl=63 time=0.471 ms
64 bytes from 10.0.2.2: icmp_req=8 ttl=63 time=0.321 ms
^C
--- 10.0.2.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7001ms
rtt min/avg/max/mdev = 0.321/0.499/0.771/0.153 ms
```

d. Verificar la comunicación remota

Hacemos ping a la dirección IP de un host remoto para comprobar si se puede comunicar más allá de un router.

Ejemplo

Tomemos una dirección ubicada en otra red como la dirección del DNS de Google 8.8.8.8 en Internet:

```
# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=63 time=40.9 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=63 time=41.0 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=63 time=40.3 ms
64 bytes from 8.8.8.8: icmp_req=4 ttl=63 time=40.9 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 40.305/40.815/41.000/0.327 ms
```

2. Bloquear/desbloquear ping

El mensaje ICMP de ping puede estar bloqueado por:

- la configuración del kernel utilizando el pseudo-sistema de archivos /proc.
- la aplicación de una regla en el firewall iptables.

Por lo tanto, esto puede impedir nuestro diagnóstico.

a. Bloquear ping con /proc

El archivo /proc/sys/net/ipv4/icmp_echo_ignore_all contiene el valor 0 por defecto y, por lo tanto, el sistema acepta las tramas ECHO ICMP.

Para bloquear el ping basta con sustituir 0 por 1. La modificación no es permanente. Para que sea persistente, debemos modificar el archivo /etc/sysctl.conf.

El comando sysctl configura en caliente los parámetros del kernel, y permite ver y cargar los parámetros de /etc/sysctl.conf.

Sintaxis

```
sysctl <opciones>
```

Opciones necesarias:

Opción		Descripción
corta	larga	
-a	--all	Muestra todos los valores disponibles.
-p	--load	Carga los parámetros del archivo mencionado o del archivo por defecto /etc/sysctl.conf.

Ejemplo

Bloquear ping:

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Para hacer el valor permanente, añadimos en el archivo /etc/sysctl.conf la línea net.ipv4.icmp_echo_ignore_all = 1.

```
# echo "net.ipv4.icmp_echo_ignore_all = 1" >> /etc/sysctl.conf
```

Cargamos los parámetros en caliente:

```
# sysctl -p
net.ipv4.icmp_echo_ignore_all = 1
```

Verificamos:

```
# sysctl -a
kernel.sched_child_runs_first = 0
kernel.sched_min_granularity_ns = 1500000
kernel.sched_latency_ns = 12000000
kernel.sched_wakeup_granularity_ns = 2000000
kernel.sched_tunable_scaling = 1
kernel.sched_migration_cost = 500000

net.ipv4.icmp_echo_ignore_all = 1

...

sunrpc.tcp_fin_timeout = 15
fscache.object_max_active = 4
fscache.operation_max_active = 2
```

b. Bloquear ping con iptables

Una regla de firewall puede rechazar un ping. Para manipular sus reglas, utilizamos el comando iptables.

Sintaxis

```
iptables <opciones>
```

Opciones:

Opción		Descripción
corta	larga	
-A	--append	Añadir al final de la cadena mencionada una regla. Se pueden utilizar tres cadenas: • INPUT para analizar los paquetes de entrada. • FORWARD para analizar los paquetes que transitan entre dos interfaces de red. • INPUT para analizar los paquetes de salida.
-D	--delete	Borra en la cadena una o varias de las reglas.
-p	--protocol	Especifica el protocolo a analizar: ICMP, IGMP, TCP, UDP...
-j	--jump	Especifica la acción a realizar si el paquete corresponde a la regla: • REJECT permite rechazar el paquete devolviendo al emisor un mensaje de error si la regla se cumple. • DROP permite rechazar el paquete sin devolver al emisor un mensaje de error si la regla se cumple. • LOG permite mostrar el resultado en la salida estándar. • ACCEPT permite aceptar el paquete si la regla se cumple.
-L	--list	Enumera todas las reglas de la cadena mencionada.
-F	--flush	Borra todas las reglas de la cadena mencionada.

Cabe señalar que desde la versión 8.04 Ubuntu ofrece UFW, una subcapa software, aunque menos completa, que permite controlar iptables de forma sencilla.

Ejemplo con Debian

Mostrar las reglas del firewall:

```
# 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
```

Comprobamos que no existen reglas definidas. Por lo tanto, se puede hacer ping al equipo.

Añadimos la regla siguiente que bloqueará el ping:

```
# iptables -A INPUT -p icmp -j DROP
```

Verificamos que está añadida:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source      destination
DROP       icmp -- anywhere  anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source      destination
```

Para volver al estado anterior, eliminamos la regla:

```
# iptables -D INPUT -p icmp -j DROP
```


Ejemplo con CentOS

Mostrar las reglas del firewall:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source     destination
ACCEPT     all  --  anywhere   anywhere    state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere   anywhere
ACCEPT     all  --  anywhere   anywhere
ACCEPT     tcp  --  anywhere   anywhere    state NEW tcp dpt:ssh
REJECT     all  --  anywhere   anywhere    reject-with icmp-host-
prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source     destination
REJECT     all  --  anywhere   anywhere    reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source     destination
```

Eliminamos la regla para el protocolo ICMP:

```
# iptables -D INPUT -p icmp -j ACCEPT
```

Verificamos:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source     destination
ACCEPT     all  --  anywhere   anywhere    state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere   anywhere
ACCEPT     tcp  --  anywhere   anywhere    state NEW tcp dpt:ssh
REJECT     all  --  anywhere   anywhere    reject-with icmp-host-
prohibited
```

```
Chain FORWARD (policy ACCEPT)
target     prot opt source      destination
REJECT     all  --  anywhere    anywhere    reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source      destination
```

Al no encontrarse ya la regla, no es posible hacer ping al equipo.

Para volver al estado anterior, reiniciamos el servicio iptables:

```
# service iptables restart
Redirecting to /bin/systemctl restart iptables.service
```

3. Puertos abiertos

Podemos comprobar los puertos TCP y UDP en escucha con los comandos netstat y ss.

Sintaxis

```
netstat <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-t	--tcp	Muestra los puertos TCP.
-u	--udp	Muestra los puertos UDP.
-a	--all	Muestra los servidores en escucha y las conexiones establecidas.
-p	--programs	Muestra el nombre y el PID de los procesos.
-e	--	Visualización extendida.
-n	--numeric	Muestra las direcciones en lugar de los nombres de host.

Ejemplo

Para ver si el puerto TCP 22 (SSH) está conectado:

```
# netstat -tpn
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address
State PID/Program name
tcp 0 0 10.0.2.15:34472 104.16.33.193:80
TIME_WAIT -
tcp 0 0 10.0.2.15:49703 198.252.206.25:443
ESTABLISHED 12879/firefox
tcp 0 0 10.0.2.15:34469 104.16.33.193:80
TIME_WAIT -
```

En efecto, el usuario bob está conectado desde el equipo 192.168.1.203.

Sintaxis

```
ss <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-t	--tcp	Muestra el protocolo TCP.
-u	--udp	Muestra el protocolo UDP.
-l	--listening	Muestra los sockets del servidor en escucha.
-p	--processes	Muestra el nombre y el PID de los procesos propietarios.
-n	--numeric	Muestra las direcciones en formato numérico.

Ejemplo

Muestra el puerto de escucha del programa Apache 2:

```
# ss -tulpn |grep apache2
tcp    LISTEN    0      0      :::80      :::*      users:
(("apache2",pid=4995,fd=4),("apache2",pid=4994,fd=4),
("apache2",pid=4991,fd=4))
```

4. Ancho de banda

El comando iftop muestra el uso del ancho de banda de una interfaz de comunicación determinada.

Sintaxis

```
iftop <opciones>
```

Opciones útiles:

Opción	Descripción
-i	Define la interfaz a escuchar.
-P	Muestra los puertos.
-N	Muestra los puertos en valores numéricos.
-n	No efectúa ninguna resolución de nombres de host.

iftop no se instala por defecto. Debian y Ubuntu disponen del paquete en sus respectivos repositorios. En cuanto a CentOS, hay que descargar y compilar las fuentes.

Instalación en Debian y Ubuntu

```
# apt-get install iftop
root@debian:~# apt-get install iftop
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  iftop
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 41,6 kB de archivos.
Se utilizarán 121 kB de espacio de disco adicional después de esta
operación.
Des:1 http://ftp.es.debian.org/debian/ jessie/main iftop
amd64 1.0~pre4-2 [41,6 kB]
Descargados 41,6 kB en 0s (397 kB/s)
Seleccionando el paquete iftop previamente no seleccionado.
(Leyendo la base de datos ... 248166 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../iftop_1.0~pre4-2_amd64.deb ...
Desempaquetando iftop (1.0~pre4-2) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
```

Configurando iftop (1.0~pre4-2) ...

Instalación en CentOS

Algunos paquetes son necesarios antes de la instalación de iftop:

```
# yum -y install libpcap libpcap-devel ncurses ncurses-devel
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror0.babylon.network
 * extras: mirror.ibcp.fr
 * updates: ftp.ciril.fr
El paquete 14:libpcap-1.5.3-8.el7.x86_64 ya se encuentra instalado
con su versión más reciente
El paquete ncurses-5.9-13.20130511.el7.x86_64 ya se encuentra
instalado con su versión más reciente
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete libpcap-devel.x86_64 14:1.5.3-8.el7 debe ser instalado
---> Paquete ncurses-devel.x86_64 0:5.9-13.20130511.el7 debe
ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package           Arquitectura Versión           Repositorio  Tamaño
=====
Instalando:
  libpcap-devel    x86_64      14:1.5.3-8.el7      base         117 k
  ncurses-devel    x86_64      5.9-13.20130511.el7 base         713 k

Resumen de la transacción
=====
Instalar 2 Paquetes
```

```

Tamaño total de la descarga: 830 k
Tamaño instalado: 2.3 M
Downloading packages:
(1/2): libpcap-devel-1.5.3-8.el7.x86_64.rpm
| 117 kB   00:00
(2/2): ncurses-devel-5.9-13.20130511.el7.x86_64.rpm
| 713 kB   00:00
-----
Total                               1.7 MB/s | 830 kB   00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando      : 14:libpcap-devel-1.5.3-8.el7.x86_64           1/2
  Instalando      : ncurses-devel-5.9-13.20130511.el7.x86_64     2/2
  Comprobando     : ncurses-devel-5.9-13.20130511.el7.x86_64     1/2
  Comprobando     : 14:libpcap-devel-1.5.3-8.el7.x86_64           2/2

Instalado:
  libpcap-devel.x86_64 14:1.5.3-8.el7
  ncurses-devel.x86_64 0:5.9-13.20130511.el7

¡Listo!

```

Descargamos las fuentes desde Internet:

```

# wget http://www.ex-parrot.com/pdw/iftop/download/iftop-0.17.tar.gz
--2016-05-05 14:39:42-- http://www.ex-
parrot.com/pdw/iftop/download/iftop-0.17.tar.gz
Resolviendo www.ex-parrot.com (www.ex-parrot.com)... 93.93.130.166,
2a00:1098:0:82:1000:0:1:2
Conectando con www.ex-parrot.com (www.ex-parrot.com)

[93.93.130.166]:80... conectado.

```

Petición HTTP enviada, esperando respuesta... 200 OK

Longitud: 160381 (157K) [application/gzip]

Grabando a: «iftop-0.17.tar.gz»

100%[=====>] 160.381 --.-K/s en 0,1s

2016-05-05 14:39:42 (1,05 MB/s) - "iftop-0.17.tar.gz" guardado

[160381/160381]

Descomprimos el archivo:

```
# tar xzvf iftop-0.17.tar.gz
```

```
iftop-0.17/
```

```
iftop-0.17/aclocal.m4
```

```
iftop-0.17/addr_hash.c
```

```
iftop-0.17/addr_hash.h
```

```
iftop-0.17/addr_dlpi.c
```

```
iftop-0.17/addr_ioctl.c
```

```
iftop-0.17/AUTHORS
```

```
iftop-0.17/bootstrap
```

```
iftop-0.17/cfgfile.c
```

```
...
```

```
iftop-0.17/util.c
```

```
iftop-0.17/vector.c
```

```
iftop-0.17/vector.h
```

Entramos en la carpeta:

<https://dogramcode.com/libros-sistemas>


```
# cd iftop-0.17
```

Compilamos los fuentes:

```
# ./configure && make && make install
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets ${MAKE}... yes
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files...o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for egrep... 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

...
```

```

/bin/sh ./config/mkinstalldirs /usr/local/sbin
  /usr/bin/install -c iftop /usr/local/sbin/iftop
/bin/sh ./config/mkinstalldirs /usr/local/man/man8
mkdir -p -- /usr/local/man/man8
  /usr/bin/install -c -m 644 ./iftop.8 /usr/local/man/man8/iftop.8
make[2]: se sale del directorio `/root/iftop-0.17'
make[1]: se sale del directorio `/root/iftop-0.17'

```

Compruebe la ejecución del binario:

```

# iftop -h
iftop: display bandwidth usage on an interface by host

Synopsis: iftop -h | [-npbBP] [-i interface] [-f filter code] [-N
net/mask]

    -h                display this message
    -n                don't do hostname lookups
    -N                don't convert port numbers to services
    -p                run in promiscuous mode (show traffic between
                     other hosts on the same network segment)
    -b                don't display a bar graph of traffic
    -B                Display bandwidth in bytes
    -i interface      listen on named interface
    -f filter code    use filter code to select packets to count
                     (default: none, but only IP packets are counted)
    -F net/mask       show traffic flows in/out of network
    -P                show ports as well as hosts
    -m limit          sets the upper limit for the bandwidth scale
    -c config file    specifies an alternative configuration file

iftop, version 0.17
copyright (c) 2002 Paul Warren <pdw@ex-parrot.com> and contributors

```

Ejemplo

```
# iftop -i eth0 -Pnn
```

	12,5Kb	25,0Kb	37,5Kb	50,0Kb	62,5Kb			
192.168.1.255:1341	=>	192.168.1.8:57501	0b	0b	0b			
	<=		0b	109b	109b			
192.168.1.255:19375	=>	192.168.1.8:19375	0b	0b	0b			
	<=		520b	104b	104b			
192.168.1.14	=>	192.168.1.8	0b	0b	17b			
	<=		0b	0b	0b			
192.168.1.14:2054	=>	192.168.1.8:49831	0b	0b	0b			
	<=		0b	0b	11b			
TX:	cum:	84B	peak:	336b	rates:	0b	0b	17b
RX:		1,22KB		520b		520b	213b	224b
TOTAL:		1,30KB		832b		520b	213b	241b

<https://dogramcode.com/libros-sistemas>

Ejercicios

1. Recopilar la configuración de una tarjeta de red

Necesitamos dos máquinas virtuales:

- la primera es una máquina Debian o Ubuntu Server.
- la segunda es una máquina CentOS.

1. Utilice la primera. Inicie una sesión como usuario root.

2. Identifique los siguientes datos:

Elemento	Valor	Comando utilizado
Nombre de la tarjeta		
Nombre del fabricante y modelo		
Controlador de la tarjeta		
Dúplex		
Velocidad de la tarjeta		
Dirección MAC		
Dirección estática o dinámica		
Dirección IP y máscara		
Gateway por defecto		
Nombre del equipo		

2. Renombrar la interfaz de red eth0

1. Utilice la segunda máquina virtual.
2. Identifique el nombre de la tarjeta de red cableada. Luego cambie el nombre a eth0. Describa su procedimiento.

Introducción

Antes de optimizar, es preciso realizar un buen análisis para comprender el problema. La lectura de los registros como `/var/log/syslog` y `/var/log/messages` es un buen comienzo para identificar entradas inusuales. Debemos asegurarnos también de que el software está actualizado de forma correcta.

Configurar el sistema operativo de forma minuciosa no resuelve todos los problemas. También podemos encontrar aplicaciones mal programadas o un hardware defectuoso. Para conducir nuestra investigación, se requiere:

- identificar el cuello de botella,
- identificar la causa,
- definir las posibles soluciones,
- implementar una solución,
- evaluar la solución.

Cuello de botella

Un cuello de botella es un componente del sistema que limita el rendimiento global. Esto puede ocultar otro problema.

Los servidores deben ser eficaces y fiables en explotación. Para optimizar sus resultados, debemos recoger datos que permitan identificar los cuellos de botella del sistema y crear un marco de referencia.

Para concebir un marco de referencia, debemos:

- identificar los recursos,
- capturar datos,
- almacenar esos datos.

1. Identificar los recursos

Se trata de elaborar un inventario de hardware y software lo más exacto posible. Con la ayuda de los comandos `lshw`, `lspci`, `lsusb`, `lscscsi`, `lsusb` entre otros... Estos últimos se han mencionado en los capítulos anteriores.

Recursos hardware:

Hardware	Comandos sugeridos
Número de CPU Velocidad de CPU	<code>lshw -C cpu</code>
Memoria RAM	<code>cat /proc/meminfo</code> <code>free -t</code>
Almacenamiento interno Almacenamiento externo	<code>lshw -C disk</code>
Interfaces de red	<code>lshw -C network</code>

Recursos software:

Sistema	Comandos sugeridos
Distribución Linux Versión de la distribución	cat /etc/*release* lsb_release -a
Kernel	uname -r
Lista de los módulos cargados	lsmod
Particionado	cat /proc/partitions lsblk fdisk -l
Versión del shell	echo \$BASH_VERSION
Lista de los procesos	ps -ef
Configuración del sistema	Archivos en /etc
Configuración de red	Archivos en /etc
Configuración de las aplicaciones	Varian según la aplicación

La lista de comandos anterior puede por su puesto evolucionar.

Entre todos esos recursos, los cuatro recursos críticos de un sistema operativo son:

- la memoria (RAM, swap),
- el/los procesador(es),
- los discos lógicos y físicos,
- la interfaz de red.

2. Capturar y almacenar los datos

La supervisión de un servidor puede efectuarse de dos maneras:

- Ad hoc, midiendo en el momento los recursos críticos del sistema.
- Durante un tiempo en el que recopilamos los datos que luego habrá que analizar.

Ad hoc

Disponemos de dos tipos de comandos:

- Los que capturan en el momento valores y los muestran. Estos valores no cambian. Es necesario volver a ejecutar el comando para actualizarlos.
- Los que capturan en el momento valores y los actualizan según un intervalo.

Durante un tiempo

Consiste en almacenar muestras de valores de un recurso cada hora de forma aproximada durante un período determinado (una semana por ejemplo). Es muy importante recoger estos datos durante fases de actividad elevada, normal y baja.

Procesador

La mayor parte de las actividades en ejecución de un equipo recurren al procesador. Un servidor puede tener uno o varios roles:

- servidor de autenticación (NIS, LDAP...),
- servidor de servicios de red (DHCP, DNS...),
- servidor de aplicación (PostgreSQL, MariaDB...),
- servidor de archivos (Samba, NFS...) o de impresión.

La actividad normal de un procesador depende del rol que se le atribuye. Un servidor de aplicaciones recurre más al procesador y a la memoria RAM para el tratamiento de datos, mientras que un servidor de archivos utiliza en primer lugar una interfaz de red y acceso a los discos.

Si combinamos varios roles, corremos el riesgo de tener los cuatro recursos críticos demasiado solicitados.

Las principales causas de congestión del procesador son los programas que toman demasiado tiempo de tratamiento y recursos de disco o red que generan demasiadas interrupciones.

El comando `top` permite ver, ad hoc, la actividad del ordenador y, lo que nos concierne en este momento, la del procesador.

Sintaxis

```
top <opciones>
```


Opciones útiles:

Opción	Descripción
-d <segundos>	Especifica el tiempo de refresco. El valor por defecto es de 3 segundos.
-u <nombre_usuario>	Muestra los procesos del usuario solicitado.

```
top - 15:46:43 up 16 min, 2 users, load average: 2,02, 0,85, 0,49
Tasks: 159 total, 1 running, 158 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,2 us, 0,3 sy, 0,0 ni, 95,7 id, 2,8 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 1533880 total, 863452 used, 670428 free, 119376 buffers
KiB Swap: 499708 total, 0 used, 499708 free. 300172 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2841	bob	20	0	1807864	212032	69288	S	1,7	13,8	0:07.11	gnome-shell
7	root	20	0	0	0	0	S	0,3	0,0	0:00.64	rcu_sched
1799	root	20	0	228520	35592	18372	S	0,3	2,3	0:02.02	Xorg
1	root	20	0	111240	5272	3092	S	0,0	0,3	0:01.76	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.03	ksoftirqd/0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:+
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/1
11	root	20	0	0	0	0	S	0,0	0,0	0:00.02	ksoftirqd/1
13	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:+
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	netns
18	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	perf
21	root	20	0	0	0	0	S	0,0	0,0	0:00.08	kworker/u4+
288	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/u4+
373	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	writeback
375	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	crypto
376	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	bioset
377	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kblockd
436	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	ata_sff
456	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	md
467	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	cfg80211
559	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	rpciod
586	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kswapd0
659	root	20	0	0	0	0	S	0,0	0,0	0:00.00	fsnotify_m+
675	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	nfsiod
725	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	acpi_therm+

En la tercera línea de la pantalla, tenemos el valor %CPU(s):

Campo	Descripción
us	Tiempo de CPU utilizado por los procesos de usuario (us de user).
sy	Tiempo de CPU utilizado por los procesos del kernel de Linux (sy de system).
ni	Tiempo de CPU utilizado por los procesos del usuario que han visto su nivel de prioridad modificado (ni de nice).
id	No empleado (id de idle).
wa	Tiempo de CPU utilizado para esperas de E/S (wa de wait o I/O-wait).
hi	Tiempo de CPU utilizado para las interrupciones hardware (hi de hardware interruptions).
si	Tiempo de CPU utilizado para las interrupciones software (si de software interruptions).
st	Tiempo de CPU de la máquina virtual actual «robado» por el hipervisor (st de stolen).

Las líneas mostradas por el comando top representan los procesos. Las columnas devuelven los siguientes datos:

Columna	Descripción
PID	PID del proceso.
USER	Usuario que ejecuta el proceso.
PR	Prioridad de la tarea.
NI	Prioridad de planificación del proceso.
VIRT	Cantidad total virtual utilizada por el proceso.
RES	Cantidad de memoria física ocupada por el proceso.
SHR	Cantidad de memoria virtual total (VIRT) realmente compartida.
S	Estado del proceso: • S (sleeping) • D (uninterruptible sleep) • R (running) • Z (zombie) • T (traced o stopped)
%CPU	Carga del procesador.
%MEM	Carga de memoria.
TIME+	Tiempo total de uso del procesador desde su arranque.
COMMAND	Nombre del comando que inició el proceso.

Podemos con la columna %CPU conocer la carga del procesador para un proceso determinado y con la columna TIME+ el tiempo total de uso del procesador desde el comienzo de la ejecución del proceso.

Valor aceptable	Valor deseado	Acción a considerar
Menos del 80%	El más bajo	Buscar el proceso que monopoliza el procesador. Actualizar o bien añadir otro procesador.

Para utilizar el comando sar en Debian y Ubuntu Server, debemos instalar el paquete sysstat. CentOS, por su parte, ya tiene el paquete instalado.

Instalación de sar para Debian y Ubuntu

Con apt-get, podemos instalar el paquete sysstat:

```
# apt-get -y install sysstat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  isag
Se instalarán los siguientes paquetes NUEVOS:
  sysstat
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 286 kB de archivos.
Se utilizarán 1.007 kB de espacio de disco adicional después de
esta operación. Des:1 http://ftp.es.debian.org/debian/ jessie/
main sysstat amd64 11.0.1-1 [286 kB]
Descargados 286 kB en 0s (1.636 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete sysstat previamente no seleccionado.
(Leyendo la base de datos ... 248110 ficheros o directorios
instalados actualmente.)
Preparando para desempaquetar .../sysstat_11.0.1-1_amd64.deb ...
Desempaquetando sysstat (11.0.1-1) ...
```

Procesando disparadores para systemd (215-17+deb8u4) ...

Procesando disparadores para man-db (2.7.0.2-5) ...

Configurando sysstat (11.0.1-1) ...

Creating config file /etc/default/sysstat with new version

update-alternatives: utilizando /usr/bin/sar.sysstat para proveer

/usr/bin/sar (sar) en modo automático

Procesando disparadores para systemd (215-17+deb8u4) ...

Sintaxis

```
sar <intervalo> <contador>
```

sar recoge los valores cada n segundos especificados en <intervalo> el número de veces indicado en <contador>.

Ejemplo

sar recoge valores cada 10 segundos 5 veces:

```
# sar 10 5
```

Linux 4.3.0 (debian)	01/05/16	_x86_64_	(2 CPU)				
17:43:03	CPU	%user	%nice	%system	%iowait	%steal	%idle
17:43:13	all	1,30	0,00	0,15	0,35	0,00	98,19
17:43:23	all	0,30	0,00	0,00	0,00	0,00	99,70
17:43:33	all	1,20	0,00	0,10	0,00	0,00	98,70
17:43:43	all	0,25	0,00	0,05	0,00	0,00	99,70
17:43:53	all	1,15	0,00	0,10	0,00	0,00	98,74
Media:	all	0,84	0,00	0,08	0,07	0,00	99,01

Memorias

En el sistema operativo Linux, existen diferentes tipos de memoria:

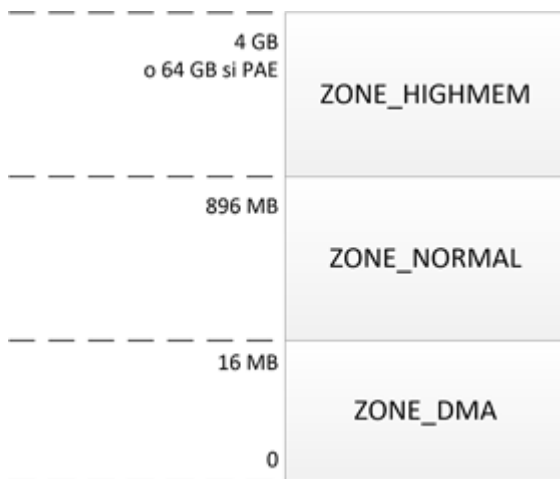
- la memoria física (physical),
- el buffer de memoria (buffers),
- la memoria caché (cached),
- el espacio de intercambio (swap),
- la memoria virtual.

1. Memoria física

La memoria física o RAM (Random Access Memory) se gestionará de manera diferente dependiendo de la arquitectura x86 32 bits o 64 bits.

a. Arquitectura x86

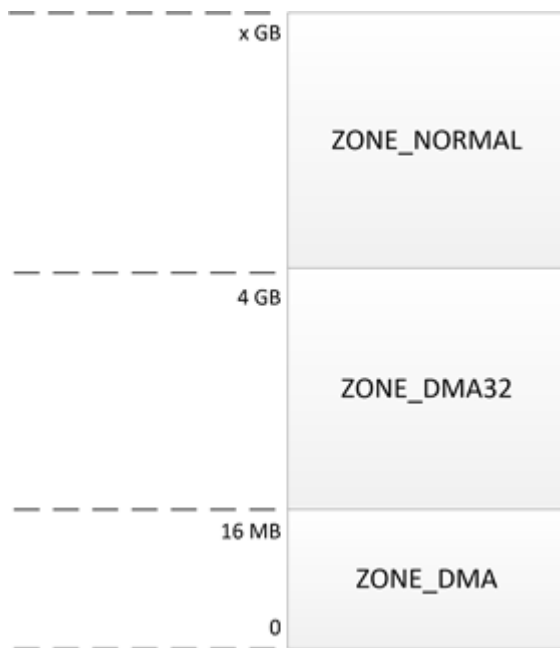
En la arquitectura x86 32 bits, la memoria física está separada en tres zonas:



La Zona ZONE_DMA corresponde a las direcciones físicas de 0 a 16 MB. La Zona ZONE_NORMAL va de 16 a 896 MB y la Zona ZONE_HIGHMEM va de 896 MB a 4 GB, e incluso 64 GB con la extensión PAE.

b. Arquitectura x86_64/amd64

La estructura de la memoria en la arquitectura x86_64 difiere de la de x86:



La zona ZONE_DMA corresponde a las direcciones físicas de 0 a 16 MB. Existe por razones históricas. La zona ZONE_DMA32 existe únicamente en las plataformas de 64 bits de 16 MB a 4 GB. La zona ZONE_NORMAL es toda la RAM disponible más allá de 4 GB.

2. Buffer

En comparación con la memoria física, un disco duro suele ser mucho más lento. Linux coloca en el buffer (buffers) los datos en espera de escritura en el disco, los datos leídos desde un disco, la posición de los bloques disponibles, etc.

El buffer permite aumentar el rendimiento del sistema operativo durante sus interacciones con dispositivos tales como los discos duros.

3. Memoria caché

Los datos almacenados en memoria caché (cached) son en su mayoría datos que rellenan la RAM en la que encontraremos en principio:

- las particiones montadas en la memoria física (tmpfs, procfs, sysfs...),
- los datos de aplicaciones terminadas,
- los datos de aplicación considerados "no esenciales" para la ejecución, como un archivo multimedia que se está reproduciendo,
- etc.

El hecho de mantener en memoria caché datos más o menos útiles permite aprovechar todo el espacio de la RAM para evitar el despilfarro.

En caso de necesidad, Linux puede liberar una zona de memoria para la ejecución de una nueva aplicación. Sin embargo, si una aplicación se ejecuta de nuevo, una parte de sus datos ya se encuentra en la memoria.

4. El espacio de intercambio

Hemos podido observar que su sistema Linux tiene en general una pequeña cantidad de datos en el swap.

En efecto, Linux no utiliza necesariamente el espacio de intercambio (swap) como memoria de seguridad. Este tipo de memoria es relativamente lento. Es interesante almacenar datos en un espacio de intercambio si no se utilizan con frecuencia.

El espacio de intercambio puede ser una partición en un disco duro o un archivo.

a. Swap como partición

La creación del espacio de intercambio como partición se realiza por lo general durante la instalación del sistema.

Si el swap es una partición primaria o una unidad lógica en una partición extendida, es más difícil cambiar el tamaño en un servidor de producción. Es preferible crear un volumen lógico.



Para más información sobre la gestión de los discos, consulte el capítulo Mantenimiento de los discos.

b. Swap como archivo

El espacio de intercambio como archivo es más lento que si se utiliza una partición. No se recomienda su uso por defecto.

Sin embargo, podemos ver que el sistema no tiene bastante swap durante una operación en concreto. Podemos aumentar su tamaño creando un archivo de intercambio. Una vez terminada la operación, podemos eliminar este último.

Ejemplo

Para crear un archivo swap de 31 MB, verificamos que estamos conectados como root.

Miramos el estado actual del swap:

```
# free -mt | grep Swap
Swap:          487          14          473
```


El equipo dispone de 487 MB de swap de los cuales 14 MB están utilizados. Quedan entonces 473 MB.

Creemos un archivo swap al que llamamos por ejemplo archivo.swp (el nombre del archivo es irrelevante):

```
# dd if=/dev/zero of=archivo.swp bs=1024 count=30000
30000+0 registros leídos
30000+0 registros escritos
30720000 bytes (512 B) copiados, 0,2433 s, 126 MB/s
```

Inicializamos archivo.swp para su uso como espacio de intercambio:

```
# mkswap archivo.swp
Configurando espacio de intercambio versión 1, tamaño = 29996 kiB
sin etiqueta, UUID=606891f9-d243-470e-9b3c-023cb4a0477b
```

Cambie los permisos del archivo para que solo pueda ser leído y modificado por el propietario root:

```
# chmod 600 archivo.swp ; ls -l archivo.swp
-rw----- 1 root root 30720000 may  6 20:34 archivo.swp
```

Active el swap:

```
# swapon -v archivo.swp
swapon archivo.swp
swapon: /root/archivo.swp: found swap signature: version 1d,
page-size 4, same byte order
swapon: /root/archivo.swp: pagesize=4096, swapspace=30720000,
devsize=30720000
```

Verificamos:

```
# free -mt | grep Swap
Swap:          517          14          502
```

El equipo dispone ahora de 517 MB de swap de los cuales 14 MB están utilizados. Quedan entonces 503 MB.

Para borrar el archivo de intercambio, es necesario en primer lugar desactivar el swap:

```
# swapoff -v archivo.swp
```

```
swapoff archivo.swp
```

Luego eliminamos archivo.swp:

```
# rm archivo.swp
```

Vuelva a mirar el estado actual del swap:

```
# free -mt | grep Swap
```

```
Swap:          487          14          473
```

El equipo dispone de nuevo de 487 MB de swap de los cuales 14 MB están utilizados. Quedan entonces 473 MB. El swap ha vuelto a su estado original.

5. Memoria virtual

La memoria virtual está compuesta por la memoria física y el espacio de intercambio. El objetivo es establecer una separación entre las direcciones manipuladas por los programas (direcciones virtuales) y las direcciones reales de los datos en memoria (direcciones físicas).

Cada programa debe disponer de un espacio de direcciones virtual incluyendo todas las direcciones virtuales. Para el correcto funcionamiento de los programas, es necesaria una correspondencia entre las direcciones virtuales y las direcciones físicas. Está garantizada por una parte por el hardware mediante un circuito adjunto a la CPU o bien integrado en la CPU, llamado MMU (Memory Management Unit) y, por otra parte a nivel software por el sistema operativo.

6. Mostrar la memoria del sistema

Para ver la memoria disponible utilice el comando free.

Sintaxis

```
free <opciones>
```

Opciones útiles:

Opción		Descripción
corta	larga	
-b	--bytes	Muestra el valor en bytes
-k	--kilo	Muestra el valor en Kbytes

-m	--mega	Muestra el valor en MB
-g	--giga	Muestra el valor en GB
--	--tera	Muestra el valor en TB.
-t	--total	Muestra el total de columnas.

Ejemplo

En una máquina de 8 GB, quedan un poco más de 6 GB disponibles y se aprecia que el swap no se utiliza.

```
# free -mt
              total        used        free      shared    buffers     cached
Mem:        7910        1684        6225          0         156         896
-/+ buffers/cache:    631        7278
Swap:      10163           0       10163
Total :    18074        1684       16389
```

La memoria utilizada por los programas es total - free - buffers - cached. La memoria utilizada como caché es la suma de buffers y cached.

El comando vmstat puede aportar también estadísticas de la memoria.

Sintaxis

```
vmstat <opción>
```

Opción necesaria:

Opción		Descripción
corta	larga	
-s	--stats	Muestra estadísticas de la memoria.

Ejemplos:

```
# vmstat -s
      8100312 K total memory
      1740980 K used memory
        747400 K active memory
        776232 K inactive memory
      6359332 K free memory
        161436 K buffer memory
        919828 K swap cache
     10407932 K total swap
           0 K used swap
     10407932 K free swap
        22948 non-nice user cpu ticks
          1 nice user cpu ticks
        7302 system cpu ticks
     4301908 idle cpu ticks
        7892 IO-wait cpu ticks
          3 IRQ cpu ticks
        2446 softirq cpu ticks
           0 stolen cpu ticks
     1022242 pages paged in
        332424 pages paged out
           0 pages swapped in
           0 pages swapped out
     1160361 interrupts
     6266581 CPU context switches
    1381810242 boot time
          4213 forks
```

El comando sar puede recoger también el estado de la memoria.

Sintaxis

```
sar <opción> <intervalo> <contador>
```

sar recopila los valores cada <intervalo> en segundos el número de veces indicado en <contador>.

Opción necesaria:

Opción	Descripción	Ejemplo
-r	Obtener el mapa de memoria del proceso.	sar recopila el mapa de memoria cada 10 segundos 2 veces:

```
root@debian:~# sar -r 10 2
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)

21:23:01 kbmemfree kbmemused %memused kbbuffers kbcached kbcommit %commit kbactive kbinact kbdirty
21:23:11 679976 860408 55,86 48248 391228 2480124 121,57 534784 226424 164
21:23:21 679916 860468 55,86 48248 391244 2480124 121,57 534712 226436 156
Media: 679946 860438 55,86 48248 391236 2480124 121,57 534748 226430 160
root@debian:~#
```

7. El mapa de memoria de un programa

Otro comando, pmap, muestra el mapa de memoria de un proceso. Es necesario conocer su PID.

Sintaxis

```
pmap <pid>
```

Opción útil:

Opción		Descripción
corta	larga	
-x	--extended	Muestra la salida en formato extendido.

Ejemplos:

Obtener el PID del shell actual con el comando ps:

```
# ps
  PID TTY          TIME CMD
 4329 pts/0    00:00:00 bash
```

4356 pts/0 00:00:00 ps

El PID es 4329. Con pmap, muestre el mapa de memoria de este proceso.

```
# pmap -x 4329
```

```
4329:  bash
```

Address	Kbytes	RSS	Dirty Mode	Mapping
0000000000400000	916	616	0 r-x--	bash
00000000006e4000	4	4	4 r----	bash
00000000006e5000	36	36	36 rw---	bash
00000000006ee000	24	24	24 rw---	[anon]
00000000009cc000	336	336	336 rw---	[anon]
00007f10cad1f000	44	16	0 r-x--	libnss_files-2.13.so
00007f10cad2a000	2044	0	0 -----	libnss_files-2.13.so
00007f10caf29000	4	4	4 r----	libnss_files-2.13.so
00007f10caf2a000	4	4	4 rw---	libnss_files-2.13.so
00007f10caf2b000	40	24	0 r-x--	libnss_nis-2.13.so
00007f10caf35000	2044	0	0 -----	libnss_nis-2.13.so
00007f10cb134000	4	4	4 r----	libnss_nis-2.13.so
00007f10cb135000	4	4	4 rw---	libnss_nis-2.13.so
00007f10cb136000	84	24	0 r-x--	libnsl-2.13.so
00007f10cb14b000	2044	0	0 -----	libnsl-2.13.so
00007f10cb34a000	4	4	4 r----	libnsl-2.13.so
00007f10cb34b000	4	4	4 rw---	libnsl-2.13.so
00007f10cb34c000	8	0	0 rw---	[anon]
00007f10cb34e000	28	16	0 r-x--	libnss_compat-2.13.so
00007f10cb355000	2044	0	0 -----	libnss_compat-2.13.so
00007f10cb554000	4	4	4 r----	libnss_compat-2.13.so
00007f10cb555000	4	4	4 rw---	libnss_compat-2.13.so
00007f10cb556000	1536	564	0 r-x--	libc-2.13.so
00007f10cb6d6000	2048	0	0 -----	libc-2.13.so
00007f10cb8d6000	16	16	16 r----	libc-2.13.so
00007f10cb8da000	4	4	4 rw---	libc-2.13.so

00007f10cb8db000	20	16	16 rw---	[anon]
00007f10cb8e0000	8	8	0 r-x--	libdl-2.13.so
00007f10cb8e2000	2048	0	0 -----	libdl-2.13.so
00007f10cbae2000	4	4	4 r----	libdl-2.13.so
00007f10cbae3000	4	4	4 rw---	libdl-2.13.so
00007f10cbae4000	148	128	0 r-x--	libtinfo.so.5.9
00007f10cbb09000	2044	0	0 -----	libtinfo.so.5.9
00007f10cbd08000	16	16	16 r----	libtinfo.so.5.9
00007f10cbd0c000	4	4	4 rw---	libtinfo.so.5.9
00007f10cbd0d000	128	116	0 r-x--	ld-2.13.so
00007f10cbd99000	1500	56	0 r----	locale-archive
00007f10cbf10000	12	12	12 rw---	[anon]
00007f10cbf23000	28	20	0 r--s-	gconv-modules.cache
00007f10cbf2a000	8	8	8 rw---	[anon]
00007f10cbf2c000	4	4	4 r----	ld-2.13.so
00007f10cbf2d000	4	4	4 rw---	ld-2.13.so
00007f10cbf2e000	4	4	4 rw---	[anon]
00007fff88204000	132	20	20 rw---	[stack]
00007fff883a5000	4	4	0 r-x--	[anon]
fffffffffff60000	4	0	0 r-x--	[anon]
-----	-----	-----	-----	
total kB	19456	2140	548	

8. Memoria insuficiente

Una memoria insuficiente es a menudo la causa de problemas de rendimiento. La búsqueda comienza por la supervisión de la memoria del servidor Linux para:

- evaluar la capacidad de memoria disponible,
- evaluar la actividad de paginación y verificar si es excesiva,
- observar los efectos de una memoria insuficiente,
- detectar una fuga de memoria.

Bajo licencia GPL, valgrind es una suite de programas informáticos para detectar problemas de gestión de memoria:

- memcheck es un detector de fugas de memoria. Ya hemos abordado este tema en el capítulo Mantenimiento de las aplicaciones.
- cachegrind es un simulador de cachés.
- callgrind es una herramienta de perfiles.

El programa sospechoso debe funcionar en el entorno de valgrind que analizará todas las E/S memoria.

Cabe recordar que hay que instalar valgrind en las distribuciones Debian y Ubuntu Server. Ya se encuentra en CentOS.

Utilización de valgrind

Sintaxis

```
valgrind <opciones> <nombre_del_programa_analizado>
```

Opciones necesarias:

Opción	Descripción
--tool	Permite seleccionar la herramienta: • memcheck • cachegrind • callgrind
--leak-check	Busca las fugas de memoria cuando el programa cliente termina. • yes y full activan la funcionalidad. • no desactiva la funcionalidad. • summary proporciona la cantidad de fugas de memoria (opción por defecto).
--leak-resolution	Define cómo se presenta el resultado de la búsqueda de fugas de memoria: • low • med • high
--show-reachable	Muestra todos los bloques a excepción de los bloques eliminados: • yes activa la funcionalidad.

	• no desactiva la funcionalidad (valor por defecto).

Ejemplo

Análisis del programa fuga2:

```
# valgrind --tool=memcheck --leak-check=full
--leak-resolution=high --show-reachable=yes ./fuga2
==3503== Memcheck, a memory error detector
==3503== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==3503== Using Valgrind-3.10.0 and LibVEX; rerun with -h for copyright
info
==3503== Command: ./fuga2
==3503==
ÉXITO - Asignación conseguida.
==3503==
==3503== HEAP SUMMARY:
==3503==      in use at exit: 20 bytes in 1 blocks
==3503==    total heap usage: 1 allocs, 0 frees, 20 bytes allocated
==3503==
==3503== 20 bytes in 1 blocks are definitely lost in loss record 1 of 1
==3503==    at 0x4C28C20: malloc (vg_replace_malloc.c:296)
==3503==    by 0x400557: main (in /home/bob/chap9/fuga2)
==3503==
==3503== LEAK SUMMARY:
==3503==    definitely lost: 20 bytes in 1 blocks
==3503==
==3503==    indirectly lost: 0 bytes in 0 blocks
==3503==      possibly lost: 0 bytes in 0 blocks
==3503==    still reachable: 0 bytes in 0 blocks
==3503==          suppressed: 0 bytes in 0 blocks
==3503==
==3503== For counts of detected and suppressed errors, rerun with: -v
==3503== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

valgrind encuentra un error de memoria.

Discos

La supervisión de los discos permite determinar la presencia de cuellos de botella:

- la necesidad de discos adicionales o más rápidos,
- un exceso de paginación.

El comando sar permite analizar la paginación del kernel.

Sintaxis

```
sar <opción> <intervalo> <contador>
```

sar recopila los valores cada <intervalo> en segundos el número de veces indicado en <contador>.

Opciones necesarias:

Opción	Descripción
-B	Recoge las estadísticas de paginación del kernel.
-d	Reporta la actividad de cada dispositivo de bloque.
-p	Muestra el nombre del dispositivo. Esta opción debe combinarse con -d.

Ejemplo 1

sar recopila la paginación de memoria cada 10 segundos 3 veces:

```
root@debian:~# sar -B 10 3
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)
22:15:14 pgpgin/s pgpgout/s fault/s majflt/s pgfree/s pgscank/s pgscand/s pgsteal/s %vmoff
22:15:24 0,00 2,40 2,50 0,00 2119,70 0,00 0,00 0,00 0,00
22:15:34 0,00 0,40 21,00 0,00 537,70 0,00 0,00 0,00 0,00
22:15:44 0,00 16,00 2,80 0,00 2011,80 0,00 0,00 0,00 0,00
Media: 0,00 6,27 8,77 0,00 1556,40 0,00 0,00 0,00 0,00
root@debian:~#
```

La columna majflt/s (major faults per second) muestra cuántas páginas se cargan desde swap a memoria. Si existe un número excesivo de majflt/s, el sistema se puede ralentizar. Esto es un indicador de una memoria insuficiente.

Ejemplo 2

Reporta la actividad de los dispositivos de bloque:

```

root@debian:~# sar -B 10 3
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)

22:15:14 ppgin/s ppgout/s fault/s majflt/s pgfree/s pgscank/s pgscand/s pgsteal/s %vmeff
22:15:24 0,00 2,40 2,50 0,00 2119,70 0,00 0,00 0,00 0,00
22:15:34 0,00 0,40 21,00 0,00 537,70 0,00 0,00 0,00 0,00
22:15:44 0,00 16,00 2,00 0,00 2011,00 0,00 0,00 0,00 0,00
Media: 0,00 6,27 8,77 0,00 1556,40 0,00 0,00 0,00 0,00
root@debian:~#
root@debian:~# sar -d -p 10 2
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)

22:18:13 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:18:23 sr0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:23 sda 0,00 0,00 6,40 8,00 0,00 0,00 0,00 0,00
22:18:23 SystemVG-rootLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:23 SystemVG-swapLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:23 SystemVG-optLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:23 SystemVG-homeLV 0,00 0,00 6,40 8,00 0,00 0,00 0,00 0,00

22:18:23 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:18:33 sr0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:33 sda 0,10 0,00 0,00 8,00 0,00 0,00 0,00 0,00
22:18:33 SystemVG-rootLV 0,10 0,00 0,00 8,00 0,00 0,00 0,00 0,00
22:18:33 SystemVG-swapLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:33 SystemVG-optLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:18:33 SystemVG-homeLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: sr0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: sda 0,45 0,00 3,60 8,00 0,00 0,00 0,00 0,00
Media: SystemVG-rootLV 0,05 0,00 0,40 8,00 0,00 0,00 0,00 0,00
Media: SystemVG-swapLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: SystemVG-optLV 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: SystemVG-homeLV 0,40 0,00 3,20 8,00 0,00 0,00 0,00 0,00
root@debian:~#

```

Si el equipo utiliza varios discos, obtendremos los mejores resultados si las solicitudes de E/S se distribuyen de forma uniforme entre todos los discos.

Los valores devueltos son las medias. Compare las columnas tps (transferencias por segundo), rd_sec/s (número de sectores leídos), y wr_sec/s (número de sectores escritos) de todos los discos.

Verifique si los valores de las columnas svctm (el tiempo medio de servicio en milisegundos para las solicitudes de E/S que han sido emitidas al dispositivo) y %util (porcentaje del tiempo de CPU durante el cual las solicitudes de E/S han sido emitidas al dispositivo) presentan valores elevados de forma constante. Una saturación del disco se produce cuando el valor de %util es cercano al 100%.

El comando iostat, que también forma parte del paquete sysstat, reporta las estadísticas de E/S para los dispositivos y las particiones.

Sintaxis

```
iostat <opciones> <dispositivo> <intervalo> <cantidad>
```

Opción útil:

Opción	Descripción
-p	Establece el dispositivo de bloque.

Ejemplos:

He aquí el resultado de un análisis global:

<https://dogramcode.com/libros-sistemas>

```

root@debian:~# iostat
Linux 3.16.0-4-amd64 (debian)    06/05/16    _x86_64_    (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,30    0,00    0,09    0,77    0,00   98,85

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdc0                0,00         0,01         0,00         78         0
sda                3,61        44,67        13,22       497285      147154
dm-0                3,54        39,24        10,80      436917      120268
dm-1                0,02         0,10         0,00        1108         0
dm-2                0,06         0,22         0,00        2399         22
dm-3                1,76         4,99         2,41       55589      26864

root@debian:~# █

```

Aquí el análisis del dispositivo /dev/sda. La pantalla incluirá 3 informes actualizados cada 3 segundos:

```

root@debian:~# iostat -p sda 3 3
Linux 3.16.0-4-amd64 (debian)    06/05/16    _x86_64_    (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,31    0,00    0,09    0,75    0,00   98,86

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                3,52        43,45        12,93       497285      147934
sda1                3,47        43,37        12,93       496369      147934

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2,85    0,00    0,17    0,00    0,00   96,98

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                0,00         0,00         0,00         0         0
sda1                0,00         0,00         0,00         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2,84    0,00    0,17    0,00    0,00   96,99

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                0,00         0,00         0,00         0         0
sda1                0,00         0,00         0,00         0         0

root@debian:~# █

```

Interfaces de red

Las principales causas de los cuellos de botella de la red son los siguientes:

- servidor sobrecargado,
- red sobrecargada,
- pérdida de integridad de la red.

El comando sar permite analizar las E/S de las interfaces de red.

Sintaxis

```
sar <opción> <palabras_clave>
```

Opción necesaria:

Opción	Descripción
-n	Recopila estadísticas de red.

Podemos utilizar palabras clave para elegir los valores a devolver. Las principales palabras clave son:

- DEV para obtener estadísticas sobre un dispositivo específico.
- EDEV para obtener estadísticas sobre los fallos de un dispositivo específico.
- ALL para generar un informe completo de las estadísticas de red.

Los valores de DEV:

Valor	Descripción
IFACE	Nombre de la interfaz.
rxpck/s	Número total de paquetes recibidos por segundo.
txpck/s	Número total de paquetes enviados por segundo.
rxkB/s	Número total de KB recibidos por segundo.
txkB/s	Número total de KB enviados por segundo.
rxcmp/s	Número total de paquetes comprimidos recibidos por segundo.
txcmp/s	Número total de paquetes comprimidos enviados por segundo.
rxmcast/s	Número total de paquetes multicast recibidos por segundo.

Ejemplo

```

root@debian:~# sar -n DEV 1 3
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)
22:39:46 IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxcst/s %ifutil
22:39:47 eth0 1,00 0,00 0,18 0,00 0,00 0,00 0,00 0,00
22:39:47 eth1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:39:47 lo 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:39:47 IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxcst/s %ifutil
22:39:48 eth0 0,99 0,00 0,06 0,00 0,00 0,00 0,00 0,00
22:39:48 eth1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:39:48 lo 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:39:48 IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxcst/s %ifutil
22:39:49 eth0 14,00 0,00 0,82 0,00 0,00 0,00 0,00 0,00
22:39:49 eth1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:39:49 lo 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxcst/s %ifutil
Media: eth0 5,32 0,00 0,35 0,00 0,00 0,00 0,00 0,00
Media: eth1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: lo 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
root@debian:~#

```

Los valores de EDEV:

Valor	Descripción
IFACE	Nombre de la interfaz.
rxerr/s	Número total de paquetes erróneos recibidos por segundo.
txerr/s	Número total de errores en la transmisión de paquetes por segundo.
coll/s	Número de colisiones durante la transmisión de paquetes por segundo.
rxdrop/s	Número de paquetes perdidos por segundo.
txdrop/s	Número de paquetes transmitidos perdidos por segundo.
txcarr/s	Número de errores de la portadora producidos por segundo.
rxfram/s	Número de errores de alineación de trama que se producen por segundo en los paquetes recibidos.
rxfifo/s	Número de errores FIFO que se producen por segundo en los paquetes recibidos.

Ejemplo

```

root@debian:~# sar -n EDEV 1 3
Linux 3.16.0-4-amd64 (debian) 06/05/16 _x86_64_ (2 CPU)

22:37:39      IFACE  rxerr/s  txerr/s  coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
22:37:40      eth0    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:40      eth1    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:40        lo    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00

22:37:40      IFACE  rxerr/s  txerr/s  coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
22:37:41      eth0    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:41      eth1    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:41        lo    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00

22:37:41      IFACE  rxerr/s  txerr/s  coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
22:37:42      eth0    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:42      eth1    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
22:37:42        lo    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00

Media:      IFACE  rxerr/s  txerr/s  coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s  txfifo/s
Media:      eth0    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
Media:      eth1    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
Media:      lo      0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
root@debian:~#

```

Ejercicio

Vamos a utilizar la máquina virtual Debian.

Inicie una sesión en línea de comandos con el usuario root.

1. Creación de un archivo swap

1. Observe el estado actual del swap y apunte los valores.
2. Cree un archivo swap de 100 MB que se llame mem.swap.
3. Active el archivo swap. Verifique que esté activo.
4. Elimine el archivo swap. Verifique que se haya eliminado.

2. Prueba de carga de memoria

1. Instale el software Stress desde los repositorios de Debian.
2. Ejecute el comando stress de la siguiente forma:

```

# stress --vm-bytes $(awk '/MemFree/{printf "%d\n", $2 * 2;}')
< /proc/meminfo)k --vm-keep -m 1 &

```

Contamos con el comando stress que simula una carga de memoria. Este funciona en segundo plano:

```

# jobs -l

```

3. Muestre con el comando free el estado de la memoria.
4. Muestre las estadísticas con vmstat.

5. Recopile la paginación con el comando `cat` cinco veces cada 10 segundos sobre la paginación.
6. Mate el proceso `stress` ejecutado en segundo plano. Verifique.
7. Muestre con el comando `free` el estado de la memoria.

Introducción

Este capítulo aborda la seguridad de un equipo Linux que forma parte de una PSSI (Política de Seguridad del Sistema de Información). En otras palabras, se trata de definir lo que está autorizado en el servidor, pero también lo que está prohibido. El método más simple es prohibir todo y luego autorizar solo lo necesario.

La seguridad de un servidor Linux se divide en dos partes:

- la seguridad física, que representa la parte del hardware;
- la seguridad lógica, que se refiere a la vez al sistema operativo y a las aplicaciones.

Los tres puntos clave para securizar un servidor son:

- el acceso, es decir, la posibilidad de acercarse al equipo para usar el teclado, o la conexión remota. Una conexión local o remota se basa en una identificación y una autenticación.
- la transmisión de flujos. El servidor Linux se comunica con otro equipo. Los datos deben estar cifrados.
- el almacenamiento de datos. El mecanismo de seguridad básico destinado a impedir el acceso a datos en un disco se basa en las ACL. Para una seguridad mayor, hay que cifrar el disco.

Seguridad física

Hay que empezar por asegurar el acceso físico al servidor, por la sencilla razón de que si la máquina física no está operativa, no contamos con sistema y, por efecto dominó, de aplicaciones.

1. ¿Quién debe tener acceso al equipo?

El acceso a un servidor debe estar autorizado en función de los datos que contiene. Solo aquellas personas autorizadas deberán entrar en la sala donde este se encuentra. La puerta debe encontrarse cerrada con llave, mediante un código o una tarjeta inteligente... Un dispositivo de vídeo vigilancia puede complementar el dispositivo.

2. Alimentación eléctrica

El servidor debe contar con una alimentación redundante para evitar problemas de funcionamiento de alimentación.

Un fallo de suministro provocaría la parada del servidor y, por lo tanto, un fallo del servicio. Para garantizar la disponibilidad del servidor debemos implementar una UPS (Uninterruptible Power Supply) que, al llega al extremo de su autonomía, envía una señal al servidor Linux para un cierre ordenado del sistema. Este hardware debe verificarse de forma periódica para garantizar su correcto funcionamiento.

Para mantener la continuidad de la red eléctrica podemos además optar por un grupo electrógeno.

3. Interfaces de comunicación

Cuando una tarjeta de red está fuera de servicio, el servidor queda indisponible.

Podemos diseñar un channel bonding que consiste en crear una interfaz de red lógica a partir de varias interfaces físicas para realizar una distribución de carga (Load Balancing) y también una tolerancia a fallos.

Los posibles modos de funcionamiento son:

Modo	Descripción	Load Balancing
0	Equilibrio de carga (round robin). Aumenta el ancho de banda y gestiona la tolerancia a fallos.	tráfico entrante
1	Modo activo-pasivo. Gestiona únicamente la tolerancia a fallos.	ninguno
2	Balanceo XOR	tráfico entrante

3	Broadcast	ninguno
4	Norma 802.3ad	ninguno
5	Balanceo TLB (Adaptive Transmit Load Balancing)	tráfico entrante
6	Balanceo ALB (Adaptive Load Balancing)	tráfico entrante y saliente

Disponemos de los siguientes parámetros para afinar la configuración:

Parámetro	Descripción	Valor
arp_interval	Este parámetro funciona con los modos 0 y 2. Define el tiempo en milisegundos entre cada petición ARP.	Por defecto 0
arp_ip_target	Define las direcciones IP (16 como máximo) si la opción arp_interval es superior a 0.	
downdelay	Define el tiempo en milisegundos para que la ausencia de una interfaz sea detectada.	Por defecto 0
lacp_rate	En el modo 802.3ad, este parámetro le permite definir el tipo de intervalo entre cada paquete LACPDU: • 0 o slow para un comando cada 30 segundos; • 1 o fast para un comando cada segundo.	Por defecto 0
max_bonds	Número máximo de agregados para esta instancia.	Por defecto 1
miimon	Indica la frecuencia de supervisión del vínculo MII (Media Independent Interface) en milisegundos. Esto determina cuántas veces debe inspeccionarse el estado del enlace de cada esclavo para detectar un posible error en la conexión. El valor recomendado es 100.	Por defecto 0

primary	Opción para los modos activo-pasivo. Determina una interfaz principal para un agregado.	
updelay	Tiempo en milisegundos para la detección de una interfaz activa.	Por defecto 0
use_carrier	Esta opción afecta a la forma en que se determina el estado del enlace. Lo que significa que sin esta opción el cambio (failover) no está activado.	Por defecto 1
xmi_hash_policy	Define la regla a emplear para determinar la interfaz para los modos balance-xor y 802.3ad. Esta opción puede tomar tres valores para determinar el algoritmo: • layer2: utiliza el XOR de la dirección MAC: (source_MAC XOR dest_MAC) módulo Num_interfaces_esclavas • layer2+3: utiliza el XOR con la dirección IP: (((source_IP XOR dest_IP) AND 0xffff) XOR (source_MAC XOR dest_MAC)) módulo Num_interfaces_esclavas • layer3+4: funciona con los protocolos TCP o UDP: ((source_port XOR dest_port) XOR ((source_IP XOR dest_IP) AND 0xffff) módulo Num_interfaces_esclavas	Por defecto layer2

Para implementar el channel bonding debemos instalar previamente el paquete ifenslave.

Debian y Ubuntu

El paquete ifenslave-2.6 debe estar instalado:

```
# apt-get install ifenslave-2.6
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  ifenslave
Se instalarán los siguientes paquetes NUEVOS:
  ifenslave ifenslave-2.6
```

```
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 21,5 kB de archivos.
Se utilizarán 33,8 kB de espacio de disco adicional después de esta
operación.
Des:1 http://ftp.es.debian.org/debian/ jessie/main
ifenslave all 2.6 [15,6 kB]
Des:2 http://ftp.es.debian.org/debian/ jessie/main
ifenslave-2.6 all 2.6 [5.940 B]
Descargados 21,5 kB en 0s (250 kB/s)
Seleccionando el paquete ifenslave previamente no seleccionado.
(Leyendo la base de datos ... 248190 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../archives/ifenslave_2.6_all.deb ...
Desempaquetando ifenslave (2.6) ...
Seleccionando el paquete ifenslave-2.6 previamente no seleccionado.
Preparando para desempaquetar .../ifenslave-2.6_2.6_all.deb ...
Desempaquetando ifenslave-2.6 (2.6) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Configurando ifenslave (2.6) ...
Configurando ifenslave-2.6 (2.6) ...
```

Cree un archivo `.conf` situado en la carpeta `/etc/modprobe.d` y asígnele el nombre `alias-bond.conf`:

```
nano /etc/modprobe.d/alias-bond.conf
```

Introduzca un alias para permitir la activación del módulo y varias opciones:

```
alias bond0 bonding
options bonding mode=0 miimon=100 downdelay=200 updelay=200
```

Cree la interfaz `bond0` en el archivo `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your
# system and how to activate them. For more information,
```

```
# see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

#Channel Bonding
auto bond0
iface bond0 inet static
    address 192.168.0.200
    netmask 255.255.255.0
    network 192.168.0.0
    gateway 192.168.0.254
    broadcast 192.168.0.255
    up /sbin/ifenslave bond0 eth0 eth1
    down /sbin/ifenslave -d bond0 eth0 eth1
```

Verifique que bond0, eth0 y eth1 tienen la misma dirección MAC en el campo HWaddr:

```
# ifconfig
bond0      Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
            inet adr:192.168.0.200  Bcast:192.168.0.255
Mask:255.255.255.0
            adr inet6: fe80::a00:27ff:fe1a:1a48/64 Scope:Link
            UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500 Metric:1
            RX packets:35933 errors:0 dropped:0 overruns:0 frame:0
            TX packets:458 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:0

            RX bytes:6727926 (6.4 KiB)  TX bytes:41766 (40.7 KiB)
```

```

eth0      Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
          inet adr:192.168.0.144  Bcast:192.168.0.255
Mask:255.255.255.0

          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:17875 errors:0 dropped:0 overruns:0 frame:0
          TX packets:325 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:3355188 (3.1 KiB)  TX bytes:30062 (29.3 KiB)


eth1      Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:18058 errors:0 dropped:0 overruns:0 frame:0
          TX packets:133 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:3372738 (3.2 KiB)  TX bytes:11704 (11.4 b)


eth2      Link encap:Ethernet  HWaddr 08:00:27:26:7B:0b
          inet adr:192.168.56.101  Bcast:192.168.56.255
Mask:255.255.255.0

          adr inet6: fe80::a00:27ff:fec9:ad0b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:396 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:8505 (8.3 KiB)  TX bytes:43632 (42.6 KiB)


lo        Link encap:Local Loopback
          inet adr:127.0.0.1  Mask:255.0.0.0
          adr inet6: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:736 errors:0 dropped:0 overruns:0 frame:0

          TX packets:736 errors:0 dropped:0 overruns:0 carrier:0

```

```
collisions:0 lg file transmisión:0  
RX bytes:55240 (53.9 KiB) TX bytes:55240 (53.9 KiB)
```

Reinicie el sistema.

CentOS

Cree un archivo .conf situado en la carpeta /etc/modprobe.d y asígnele el nombre alias-bond.conf:

```
# nano /etc/modprobe.d/alias-bond.conf
```

Introduzca un alias para permitir la activación del módulo:

```
alias bond0 bonding
```

Cree un archivo para la interfaz bond0:

```
# nano /etc/sysconfig/network-scripts/ifcfg-bond0
```

Introduzca este contenido:

```
DEVICE=bond0  
IPADDR=172.16.1.207  
NETMASK=255.255.255.0  
ONBOOT=yes  
BOOTPROTO=none  
USERCTL=no  
BONDING_OPTS="mode=0 miimon=1000"
```

Edite el archivo ifcfg-eth0 como sigue:

```
DEVICE=eth0  
ONBOOT=yes  
MASTER=bond0  
SLAVE=yes  
BOOTPROTO=none  
  
USERCTL=no  
TYPE=Ethernet
```

Cambie también el archivo ifcfg-eth1.

```
DEVICE=eth1
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
TYPE=Ethernet
USERCTL=no
```

Active la interfaz bond0:

```
#ifconfig bond0 up
```

Verifique:

```
# ifconfig
bond0      Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
            inet adr:172.16.1.207  Bcast:172.16.1.255
            Mask:255.255.255.0
            adr inet6: fe80::a00:27ff:fe61:1b3f/64 Scope:Link
            UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500 Metric:1
            RX packets:112 errors:0 dropped:0 overruns:0 frame:0
            TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:0
            RX bytes:11663 (11.3 KiB)  TX bytes:4041 (3.9 KiB)

eth0       Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
            inet adr:192.168.1.205  Bcast:192.168.1.255
            Mask:255.255.255.0
            UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
            RX packets:56 errors:0 dropped:0 overruns:0 frame:0
            TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:1000
            RX bytes:5728 (5.5 KiB)  TX bytes:3489 (3.4 KiB)

eth1       Link encap:Ethernet  HWaddr 08:00:27:61:1B:3F
```



```

UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
RX packets:56 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:5935 (5.7 KiB) TX bytes:552 (552.0 b)

eth2      Link encap:Ethernet HWaddr 08:00:27:26:7B:2F
          inet adr:192.168.56.101 Bcast:192.168.56.255
Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fe26:7b2f/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:355 errors:0 dropped:0 overruns:0 frame:0
TX packets:239 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:37580 (36.6 KiB) TX bytes:39520 (38.5 KiB)

lo        Link encap:Local Loopback
          inet adr:127.0.0.1 Mask:255.0.0.0
          adr inet6: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmisión:0
RX bytes:0 (0.0 KiB) TX bytes:0 (0.0 KiB)

```

Las interfaces bond0, eth0 y eth1 tienen la misma dirección MAC.

4. BIOS

La BIOS debe estar garantizada por una contraseña y su acceso limitado a personal autorizado para evitar la modificación de la configuración del equipo.

Ejercicio del capítulo Arquitectura del sistema GNU/Linux

El objetivo es identificar los diferentes componentes del sistema.

1. Antes de comenzar, abra una consola en una de las tres distribuciones que comentamos en este libro. Luego, identifíquese como root.

Para las distribuciones Debian y Ubuntu:

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password :
bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

Podemos efectuar esta misma operación con CentOS si contamos con la cuenta del usuario bob definida como sudoer. Para esto, es necesario que sea miembro del grupo wheel. En caso contrario, contamos con otras posibilidades:

- Iniciar una sesión como root, ya que este se encuentra desactivado.
- Iniciar una sesión con la cuenta de usuario normal bob y usar el comando su:

```
CentOS Linux 7 (core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an X86_64

srv02 login : bob
Password :
[bob@srv02 ~]$ su -
Contraseña :
[root@srv02 ~]#
```

2. Identifique, empleando diferentes comandos, los siguientes datos:

- La arquitectura física con uname:

```
# uname -m
```

```
x86_64
```

- La versión del kernel:

```
# uname -r 3.16.0-4-amd64
```

- El nombre y la versión del shell:

```
# echo $BASH_VERSION  
4.3.30(1)-release
```

- La versión de GRUB:

```
# grub-mkconfig -v  
grub-mkconfig (GRUB) 2.02~beta2-22
```

- La versión de la distribución de Linux:

```
# lsb_release -a  
No LSB modules are available.  
Distributor ID: Debian  
Description:    Debian GNU/Linux 8.2 (jessie)  
Release:        8.2  
Codename:       jessie
```

También tenemos esta solución:

```
# cat /etc/*release*  
PRETTY_NAME="Debian GNU/Linux 8 (jessie)"  
NAME="Debian GNU/Linux"  
VERSION_ID="8"  
VERSION="8 (jessie)"  
ID=debian  
HOME_URL="http://www.debian.org/"  
SUPPORT_URL="http://www.debian.org/support/"  
BUG_REPORT_URL="https://bugs.debian.org/"
```

- Los servicios arrancados:

Debian:

```
# insserv -s
K:04:0 1 6:rsyslog
K:08:0 6:umountfs
K:05:0 6:umountnfs.sh
K:03:0 6:sendsigs

*** Salida truncada ***

S:11:S:udev-finish
S:03:2 3 4 5:ssh
S:01:2 3 4 5:sudo
```

Ubuntu Server:

```
# initctl list
mountall-net stop/waiting
ountnfs-bootclean.sh start/running
passwd stop/waiting
rc stop/waiting
rsyslog start/running, process 575

*** Salida truncada ***

console-font stop/waiting
network-interface-container stop/waiting
ureadahead stop/waiting
```

CentOS:

```
# systemctl --type service
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
abrt-ccpp.service	loaded	active	exited	Install ABRT coredump hook
abrt-oops.service	loaded	active	running	ABRT kernel log watcher
abrt-xorg.service	loaded	active	running	ABRT Xorg log watcher

*** Salida truncada ***

```
tuned.service          loaded active running Dynamic System Tuning Daemon
upower.service         loaded active running Daemon for power management
```

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

65 loaded units listed. Pass --all to see loaded but inactive units, too.

To show all installed unit files use 'systemctl list-unit-files'.

Ejercicio del capítulo Kernel Linux

Vamos a utilizar la distribución de nuestra elección.

1. Inicie una sesión como root y abra una línea de comandos:

- Para las distribuciones Debian y Ubuntu:

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password :
bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

- Para la distribución CentOS:

```
CentOS Linux 7 (core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an X86_64

srv02 login : root
Password :
[root@srv02 ~]#
```

2. Descargue el último kernel de Linux en el directorio /root:

```
# wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.3.tar.xz
```

3. Descomprima el archivo y entre en el directorio Linux:

```
# tar xJvf linux-4.3.tar.xz
# cd linux-4.3.tar.xz
```

4. Asegúrese de disponer de todos los requerimientos de software:

- Para las distribuciones Debian y Ubuntu:

```
# apt-get install build-essential module-init-tools fakeroot
```

- Para la distribución CentOS:

```
# yum groupinstall "Development Tools"
```

5. Elija el archivo de configuración del kernel activo:

```
# make defconfig
```

6. Compile el kernel (método clásico) y luego instálelo:

```
# make && make modules_install install
```

7. Reinicie el sistema con el nuevo kernel:

```
# shutdown -r now
```

En el menú de GRUB, asegúrese de que el sistema arranca bien con el nuevo kernel. Inicie una sesión y muestre la versión del kernel:

```
# uname -r
4.3.0
```

Ejercicio del capítulo Módulos

1. Cree un directorio LKM en su directorio de base y entre en él.

2. Escriba el código siguiente:

<https://dogramcode.com/libros-sistemas>

```

#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("John Doe");
MODULE_DESCRIPTION("Modulo hello world");
MODULE_VERSION("Version 1.00");

int init_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función
init_module().\n");
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "[Hello world] - Se llama a la función
cleanup_module().\n");
}

```

3. Escriba el archivo makefile:

```

obj-m += hello.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD)
modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

install:

```

```
cp ./hello.ko /lib/modules/$(shell uname -r)
/kernel/drivers/misc
```

4. Compile y luego instale:

```
# make && make install
```

5. Cargue el módulo de forma automática al arrancar el sistema.

Debemos actualizar la base de datos modules.dep:

```
# depmod -a
```

Para las distribuciones Debian y Ubuntu hay que añadir al final del archivo /etc/modules:

```
# /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.

Loop
hello
```

En cuanto a la distribución CentOS, será necesario contar con el script hello.modules en la carpeta /etc/sysconfig/modules siguiente:

```
#!/bin/sh

/sbin/modprobe hello > /dev/null 2>&1
```

Cambie los permisos:

```
# chmod 755 /etc/sysconfig/modules/hello.modules
```

6. Compruebe la carga tras el reinicio del sistema:

```
# lsmod |grep hello
```

7. Muestre por pantalla la información del módulo:

```
# modinfo hello
```


8. Descargue el módulo y compruebe que el módulo hello se ha descargado de forma correcta:

```
# modprobe -rv hello; lsmod |grep hello
```

O bien:

```
# rmmod hello ; lsmod |grep hello
```

Ejercicio del capítulo Pseudo-sistemas de archivos

El objetivo es modificar de manera permanente la gestión del swap.

1. Inicie una sesión como usuario root:

- Para las distribuciones Debian y Ubuntu:

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password :
bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

- Para la distribución CentOS:

```
CentOS Linux 7 (core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an X86_64

srv02 login : root
Password :
[root@srv02 ~]#
```

2. ¿Cuál es el parámetro del kernel de Linux que controla la cantidad de datos que debe conservarse en la RAM o puesta en swap? Muestre su valor.

El parámetro del kernel es /proc/sys/vm/swappiness. Su valor es:

```
# cat /proc/sys/vm/swappiness
```

3. Cambie el valor de este parámetro a 10. Asegúrese de que este cambio sea permanente. Reinicie el sistema y verifique.

El cambio permanente se realiza en el archivo `/etc/sysctl.conf`. Agregue a este último la línea de configuración siguiente:

```
vm.swappiness = 10
```

Después de reiniciar el sistema, verifique:

```
# sysctl vm.swappiness
10
```

Ejercicio del capítulo Solución de problemas hardware

1. Inicie una sesión en una consola como el usuario root.

- Para las distribuciones Debian y Ubuntu:

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password :
bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

- Para la distribución CentOS:

```
CentOS Linux 7 (core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an X86_64

srv02 login : root
Password :
[root@srv02 ~]#
```

2. Compruebe la presencia del comando `lshw`. Si no se encuentra, instálelo.

- Debian:

```
# apt-get install lshw
```

- El comando ya se encuentra instalado en la distribución Ubuntu Server.
- CentOS no cuenta con esta herramienta en sus repositorios. Debemos descargar este paquete RPM desde Internet (los motores de búsqueda pphone.net o rpmfind.et nos pueden ayudar) y luego instalarlo:

- `# wget ftp://mirror.switch.ch/pool/4/mirror/centos/7.1.1503/os/`
- `x86_64/ packages/lshw-B.02.17-2.el7.x86_64.rpm`

```
# rpm -ivh lshw-B.02.17-2.el7.x86_64.rpm
```

Cabe señalar que la versión del paquete y la dirección del sitio web pueden evolucionar con el tiempo.

3. Identifique el nombre de la interfaz de comunicación cableada con lshw y el nombre del controlador del dispositivo:

```
# lshw -c network
```

```
*-network
```

```
description: Ethernet interface
```

```
product: 82540EM Gigabit Ethernet Controller
```

```
vendor: Intel Corporation
```

```
physical id: 3
```

```
bus info: pci@0000:00:03.0
```

```
logical name: eth0
```

```
version: 02
```

```
serial: 08:00:27:dbb:f:ad
```

```
size: 1GB/s
```

```
capacity: 1GB/s
```

```
width: 32 bits
```

```
clock: 66MHz
```

```
capabilities: pm pcix bus_master cap_list ethernet physical
```

```
tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
```

```
configuration: autonegotiation=on broadcast=yes driver=e1000
```

```
driverversion=7.3.21-k8-NAPI duplex=full firmware=N/A ip=10.0.2.15
```

```
latency=64 link=yes mingnt=255 multicast=yes port=twisted pair
speed=1GB/s
    resources: irq:19 memory:f0000000-f001ffff
ioport:d010(size=8)
```

El nombre lógico de la interfaz Ethernet 82540EM Gigabit Ethernet Controller es eth0. El controlador de dispositivo (driver), que es un módulo del kernel, es e1000.

4. Visualice las entradas del registro del controlador.

El registro donde se encuentra la información relativa a los dispositivos es /var/log/dmesg:

```
# dmesg |grep e1000
[  0.280719] e1000: Intel(R) PRO/1000 Network Driver - version
7.3.21-k8-NAPI
[  0.280720] e1000: Copyright (c) 1999-2006 Intel Corporation.
[  0.634449] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit)
08:00:27:3a:17:d8
[  0.634449] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network
Connection
[  7.108646] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow
Control: RX
```

5. Compruebe si el controlador está cargado:

```
# lsmod |grep e1000
e1000                122545  0
```

6. Descargue el módulo del kernel que controla esta interfaz.

- Con modprobe:

```
# modprobe -rv e1000
```

- Con rmmod:

```
# rmmod e1000
```

7. Compruebe con `lshw` si la interfaz de comunicación no ha dejado un nombre lógico o un controlador.

```
# lshw -c network
*-network UNCLAIMED
    description: Ethernet controller
    product: 82540EM Gigabit Ethernet Controller
    vendor: Intel Corporation
    physical id: 3
    bus info: pci@0000:00:03.0
    version: 02
    width: 32 bits
    clock: 66MHz
    capabilities: pm pcix cap_list
    configuration: latency=64 mingnt=255
    resources: memory:f0000000-f001ffff ioport:d010(size=8)
```

Observamos que la interfaz ya no cuenta con el nombre lógico `eth0` y la ausencia del piloto `e1000`.

8. Descargue el módulo del kernel que controla esta interfaz.

```
# modprobe e1000
```

Ejercicio del capítulo

Mantenimiento de los discos

1. Particionado durante la instalación

1. Cree una nueva máquina virtual:

<https://dogramcode.com/libros-sistemas>

Componentes	Valores
General	Linux Debian 8 (64 bits) 768 MB de RAM
Sistema	Orden de arranque: disco duro óptica
Almacenamiento	Controlador CD/DVD Imagen ISO DVD1 de Debian 8 Cinco discos duros (asignados dinámicamente): Disco 1: 8 GB Disco 2: 10 GB Disco 3: 5 GB Disco 4: 10 GB Disco 5: 9 GB
Red	Adaptador 1 conectado a NAT

2. Instale Debian 8 con la configuración siguiente:

Componentes	Valores
Cuenta root	Desactivada
Cuenta bob	Sudoer Contraseña: formación
Partición primaria	/boot : 1 GB (ext2)
Grupo de volúmenes	Nombre: sysVG Lista de PV: /dev/sda /dev/sdb

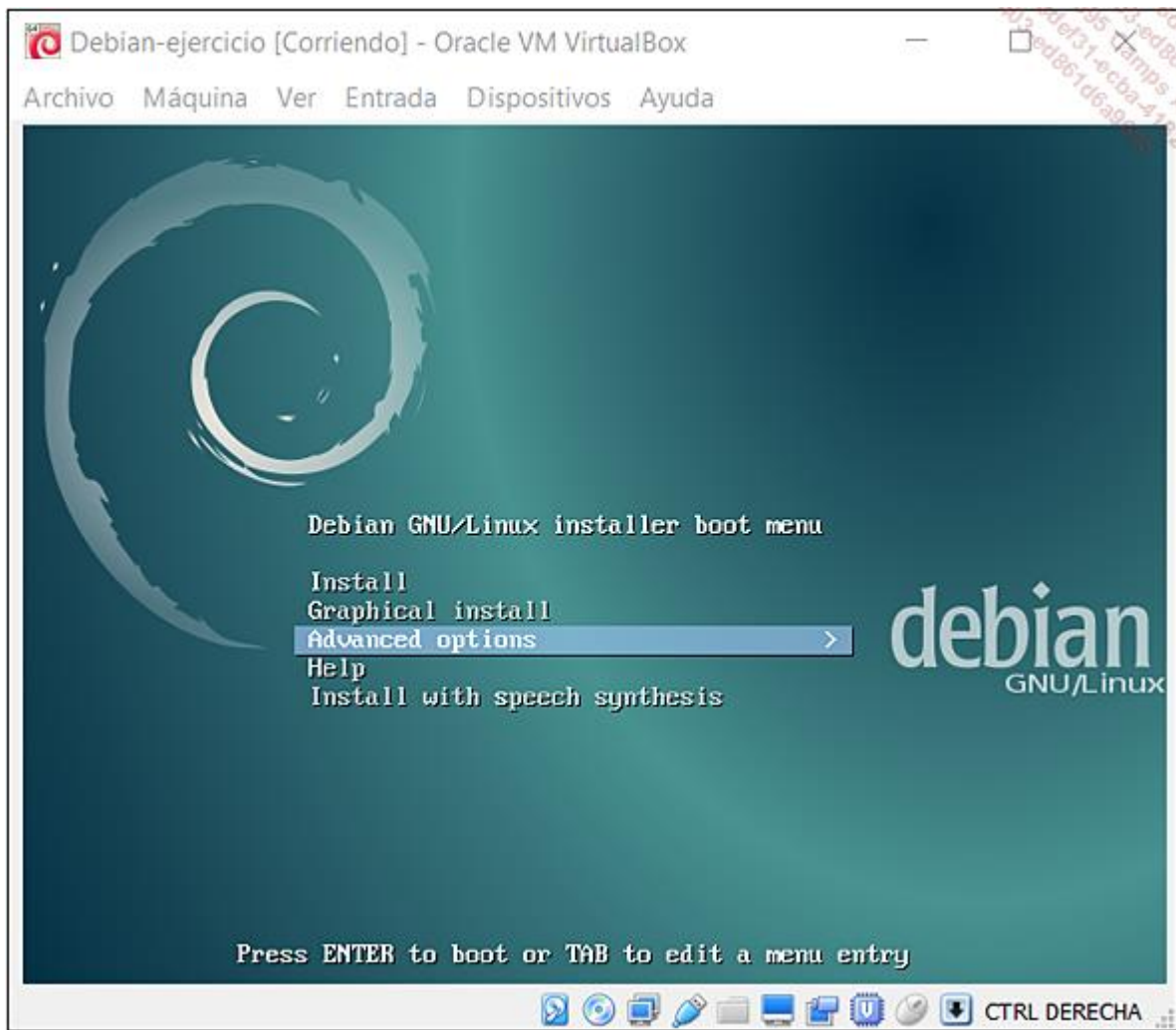
Volúmenes lógicos	Swap: swapLV de 1204 MB /: rootLV de 8 GB (ext4) /usr: rootLV de 5 GB (ext4) /var: varLV de 4 GB (ext4)
Interfaz gráfica	no
Paquetes	Por defecto
Red	Cliente DHCP

3. Inicie una sesión como bob y visualice los PV, el VG y los LV.

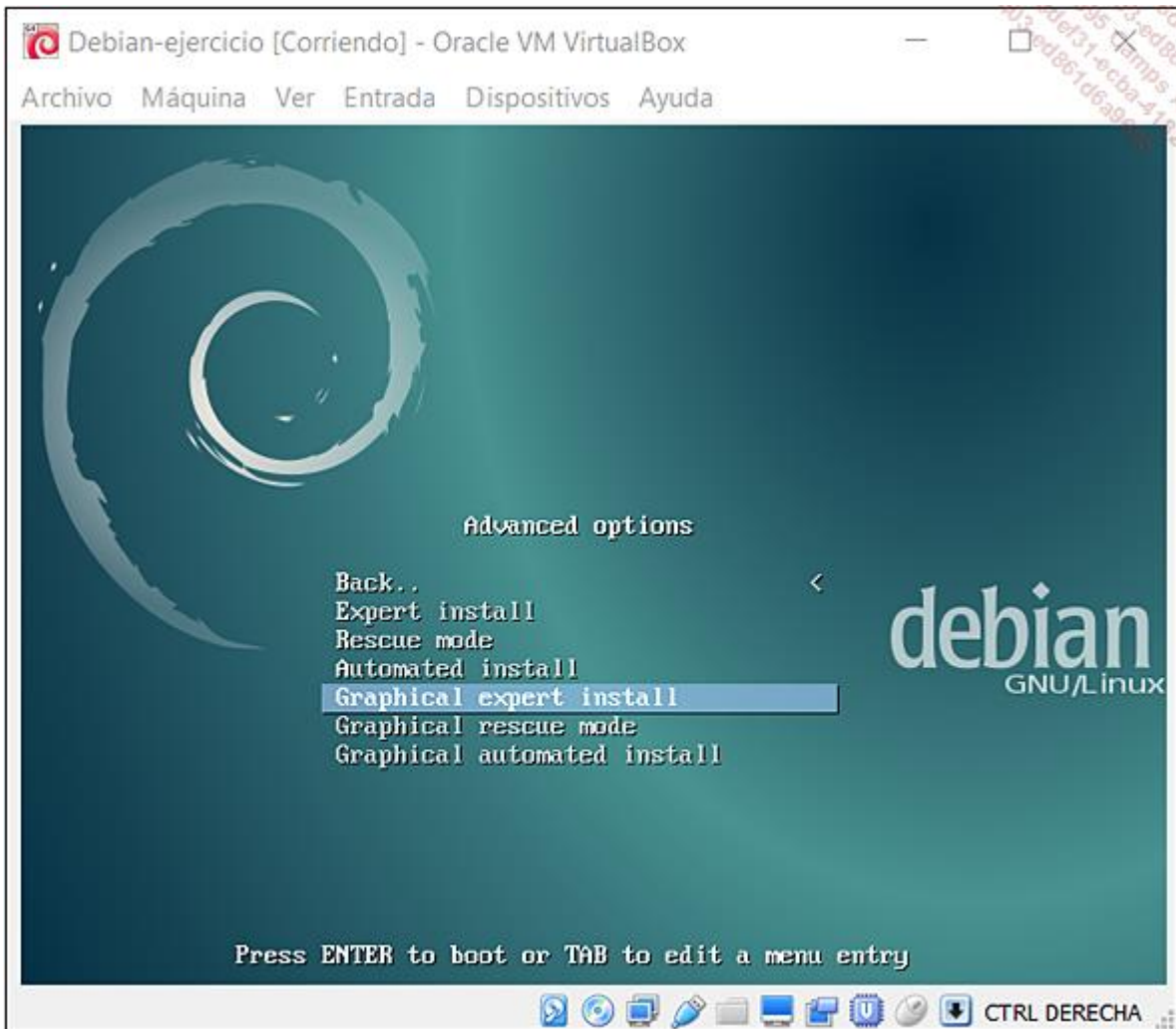
Una vez creada la máquina virtual en VirtualBox, la iniciamos.

El libro no incluye todas las capturas de pantalla de la instalación, solo los pasos clave. Se aconseja contar con una conexión a Internet.

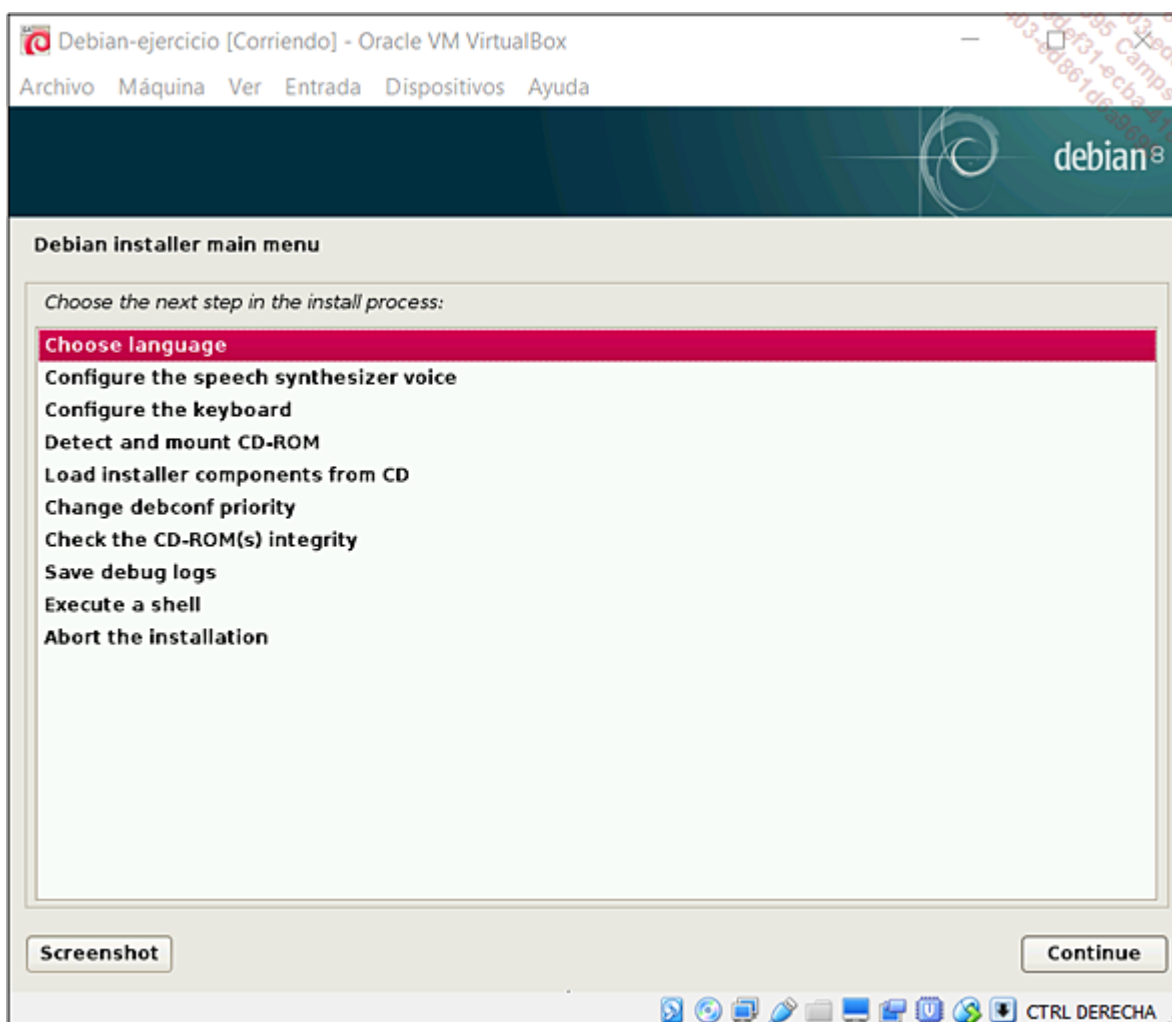
Seleccione en el menú **Advanced options**:



Luego **Graphical expert install**:



Aparece el primer menú de instalación:



Seleccione el idioma Español, el país España, los parámetros regionales es_ES.UTF-8 y el teclado español. Puede elegir otro idioma y otro país si fuera necesario.

No se necesitan componentes adicionales a instalar.

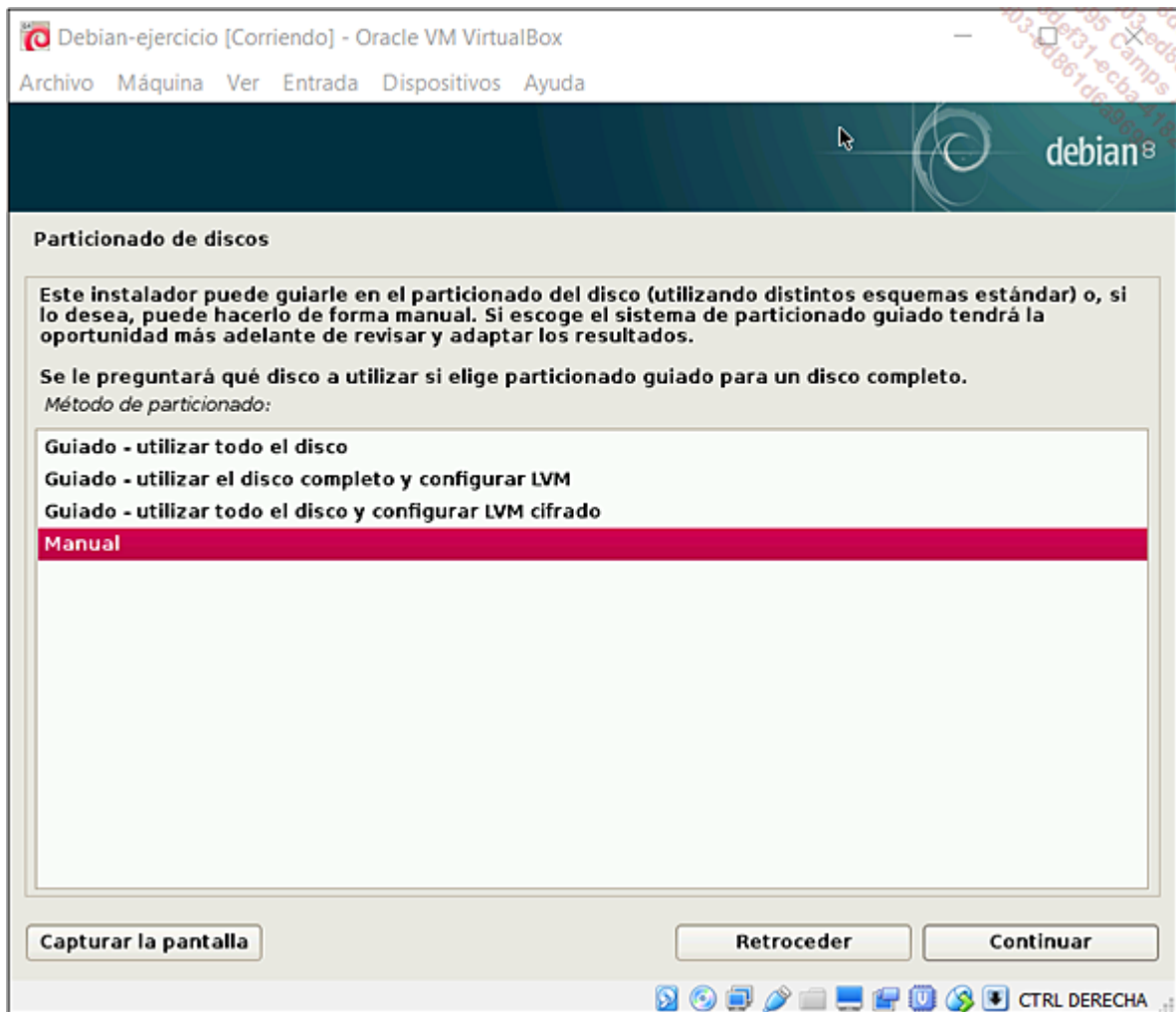
La configuración de red se efectuará automáticamente (DHCP).

El nombre del equipo será debsrv y el sufijo DNS staff.local. También podrá configurar otros valores.

Las contraseñas deben estar en el archivo /etc/shadow. Rechazamos la conexión de root. Así que crearemos un usuario que será sudoer. El nombre de inicio de sesión será bob. Su contraseña será: formación.

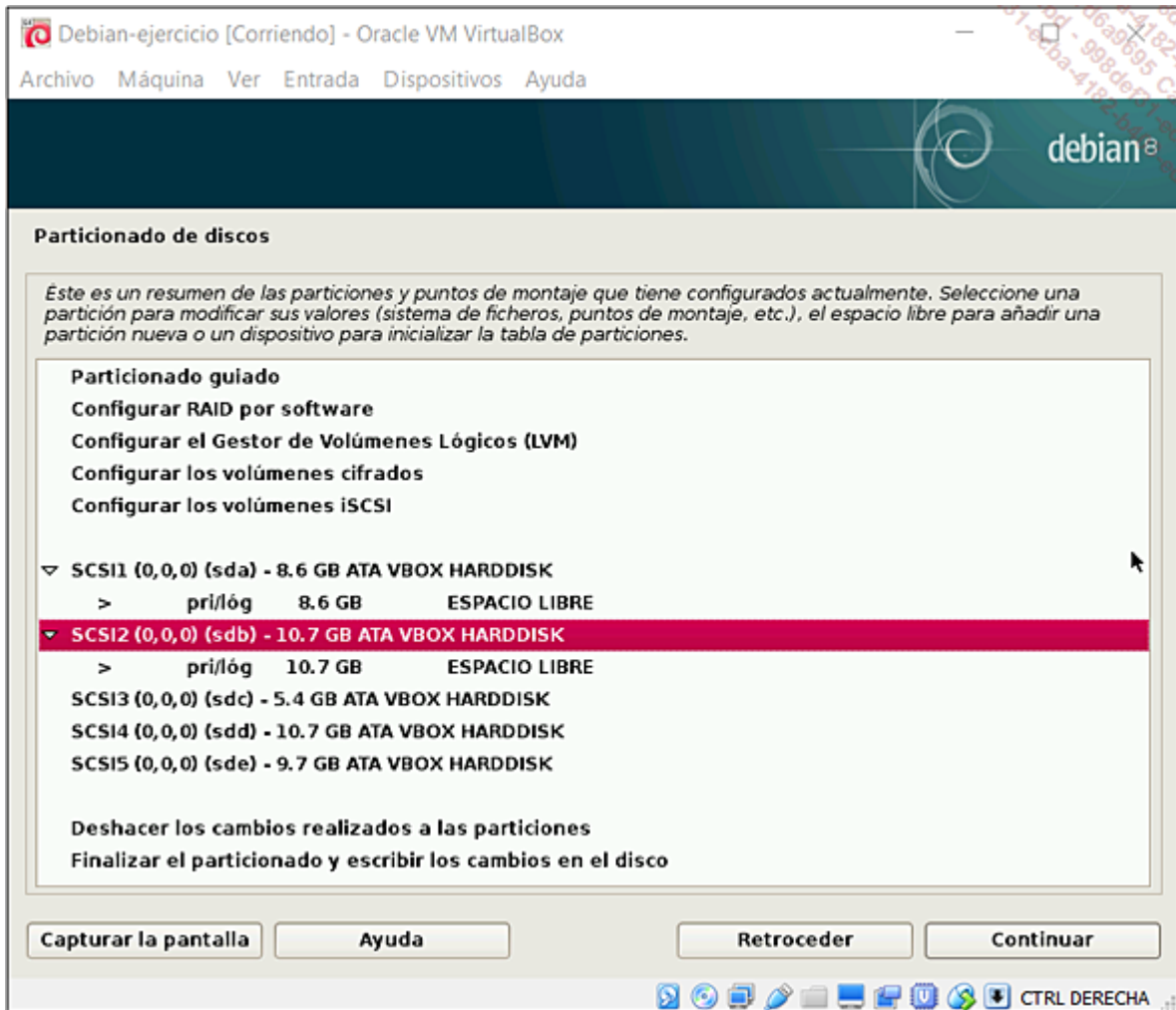
Acepte la configuración del reloj. Acepte el servidor de tiempo propuesto 0.debian.pool.ntp.org - Península. Puede adaptarlo si se encuentra fuera de España.

Pasamos a la detección de los discos. Vamos a elegir el particionado manual.



Seleccione el disco de 8 GB /dev/sda y cree una tabla de partición msdos (MBR).

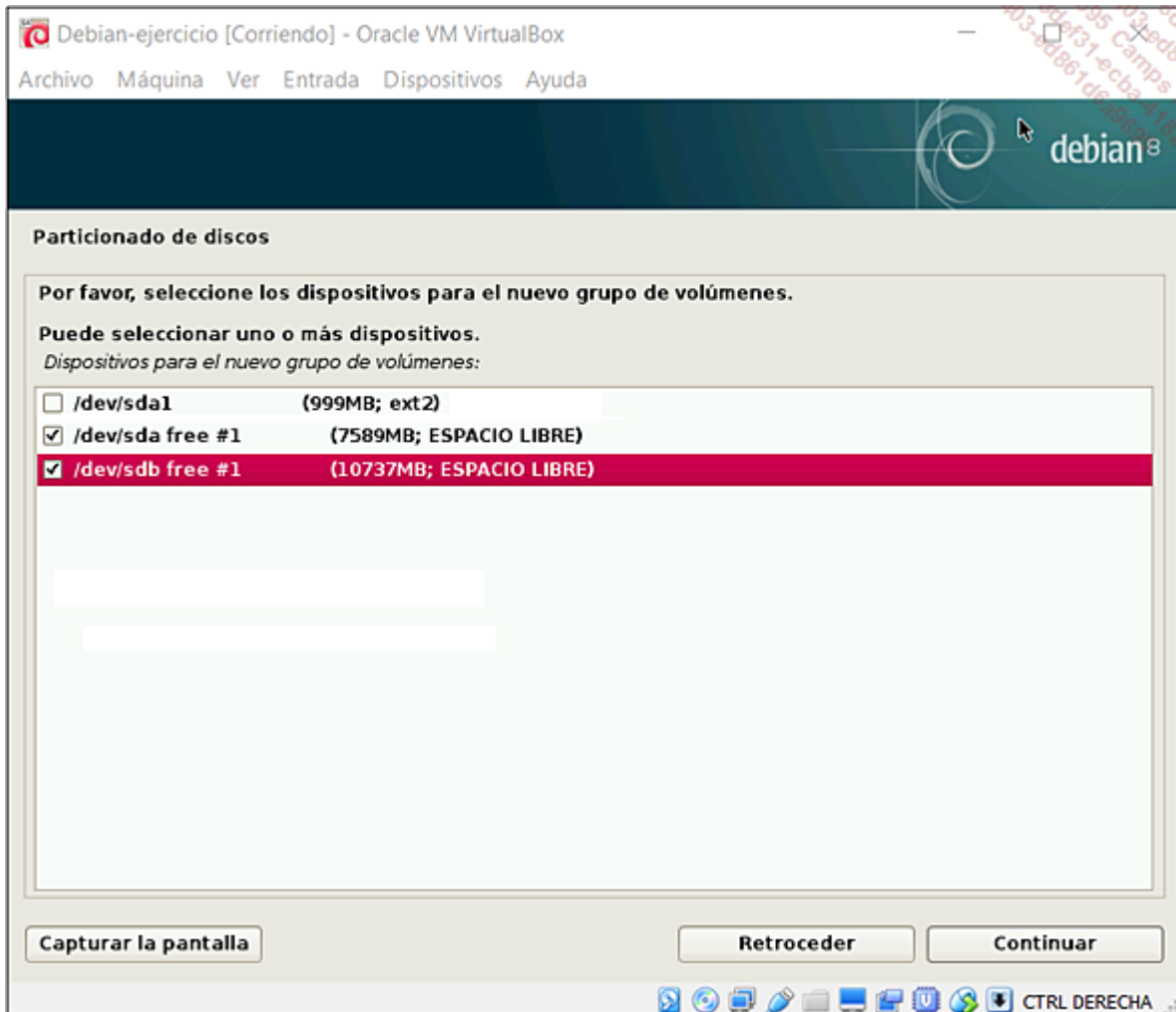
Seleccione el disco de 10 GB /dev/sdb y cree una tabla de partición msdos (MBR).



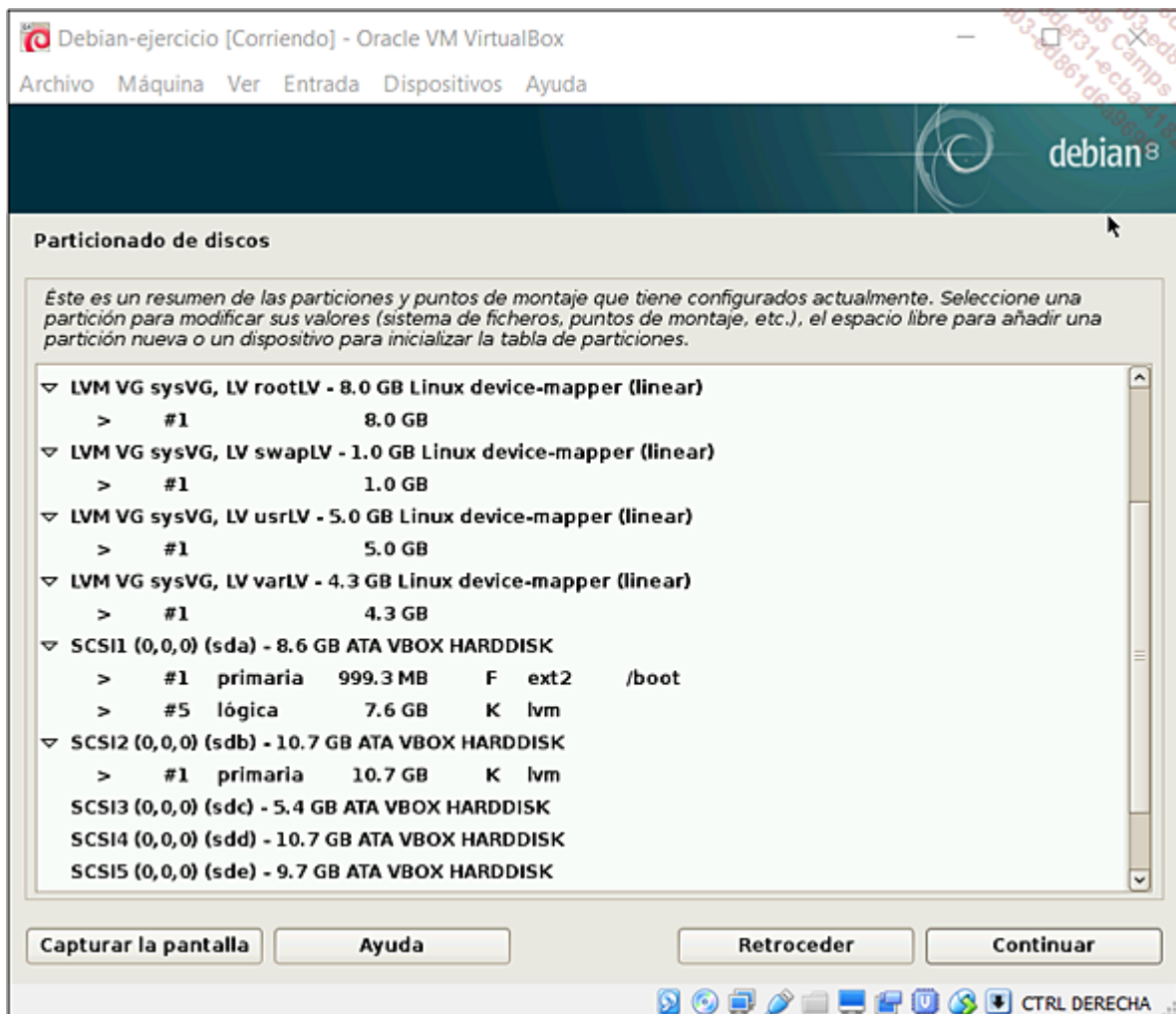
Deje los otros tres discos (sdc, sdd y sde) tal como están.

En el disco /dev/sda, cree una partición primaria de 1 GB que habrá que formatear en ext2.
Su punto de montaje será /boot.

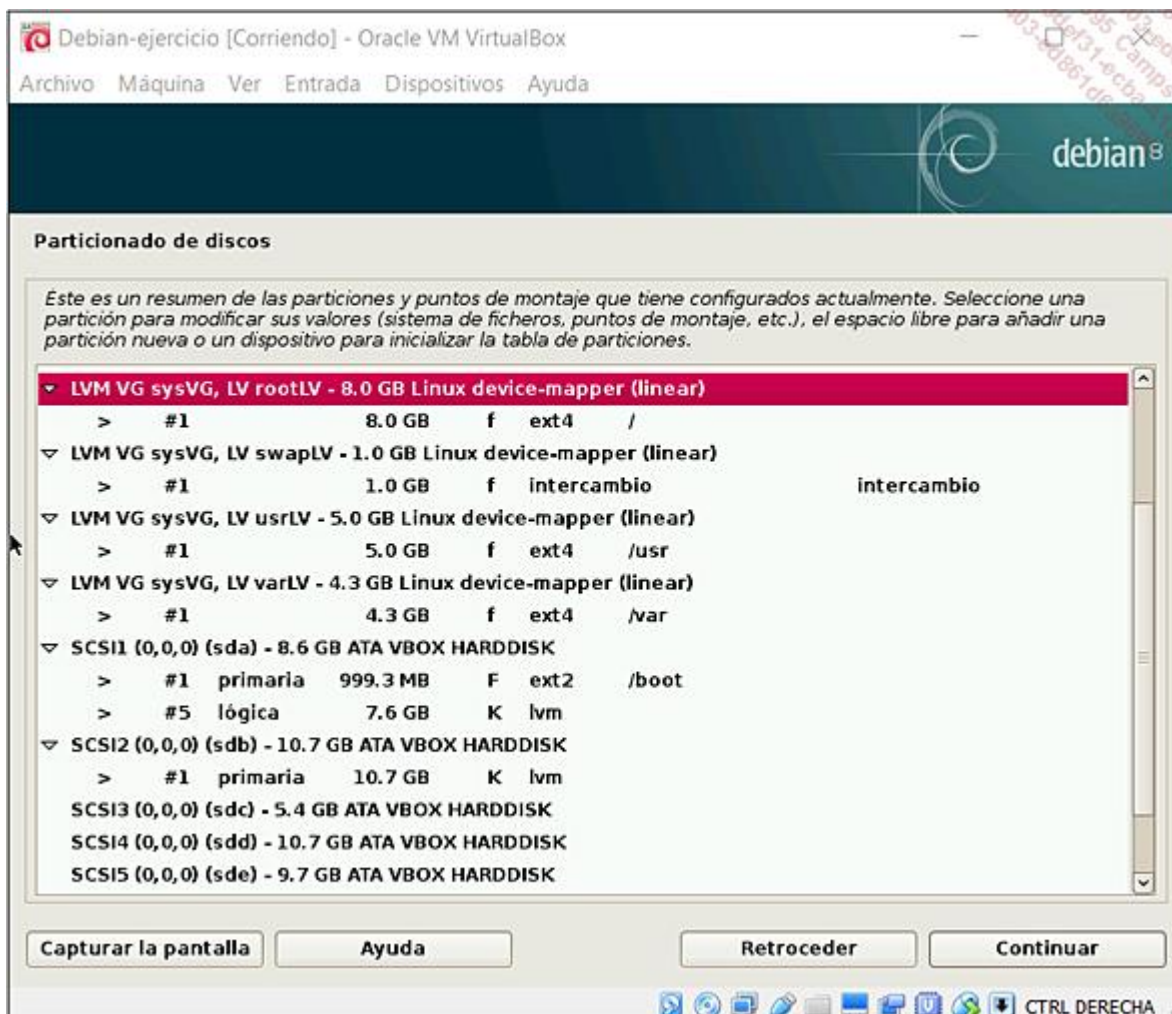
Cree un VG sysVG con los dos PV /dev/sda y /dev/sdb:



El VG sysVG tiene 18 GB aproximadamente. Cree los cuatro LV swapLV de 1204 MB, rootLV de 8 GB (ext4), usrLV de 5 GB (ext4) y varLV de 4 GB (ext4):



Seleccione el sistema de archivos swapFS para swapLV y ext4 para los otros tres LV, y defina sus puntos de montaje respectivos: /root para rootLV, /usr para usrLV y /var para varLV:



Valide la configuración de los discos para continuar con la instalación.

Instale el sistema base. Seleccione la imagen del kernel: linux-image-3.16.0-4-amd64. Tome solo los controladores necesarios para este sistema.

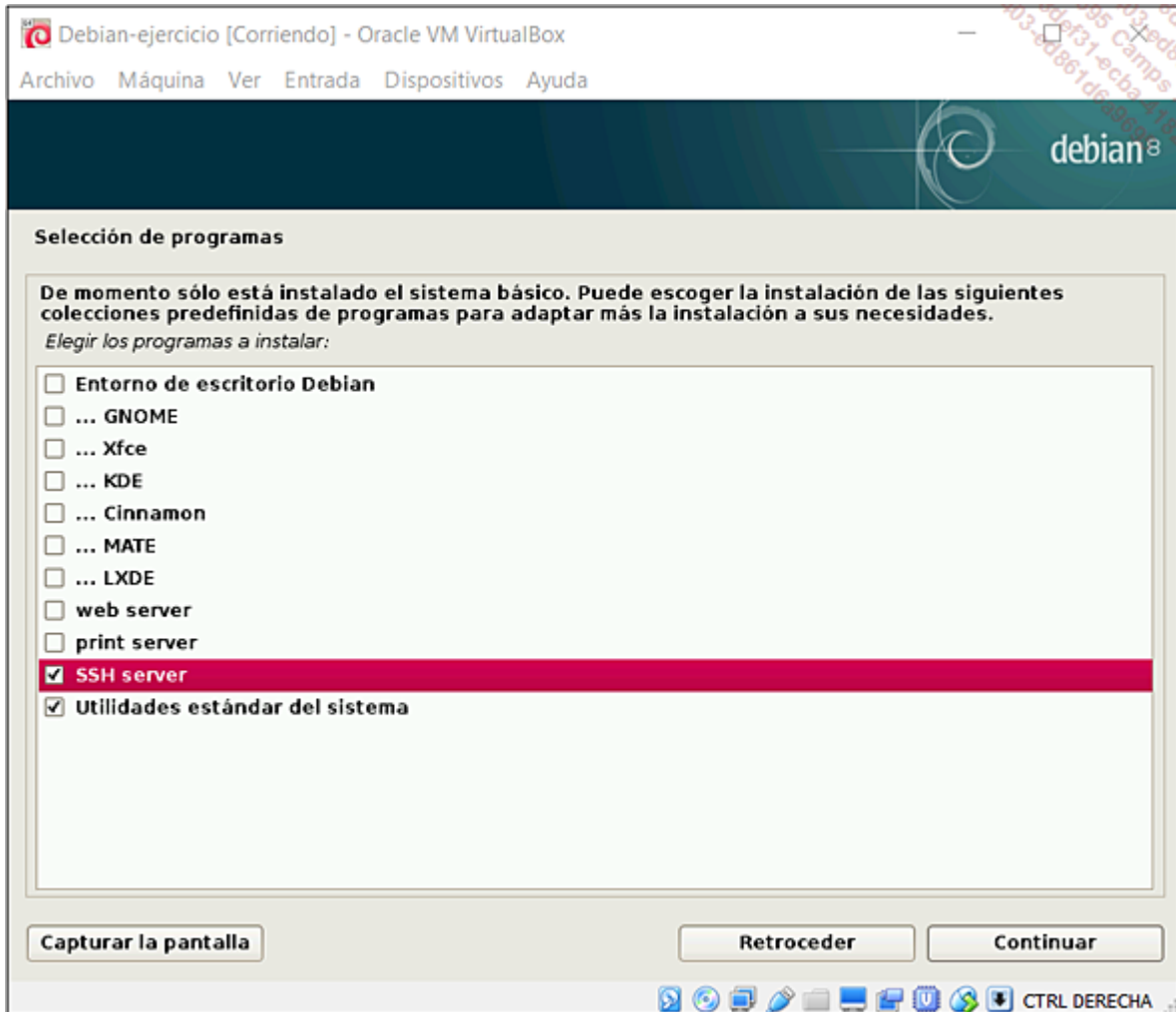
Rechace la configuración de paquetes con otros CD. Tome un servidor espejo HTTP, por ejemplo ftp.es-debian.org. No se necesita servidor proxy.

Puede aceptar el uso de software no libre, así como las actualizaciones de seguridad y de publicación.

Pasamos a la selección e instalación del software.

Decline participar en el estudio estadístico sobre el uso de los paquetes.

Seleccione solo el servidor SSH y las utilidades habituales del sistema:



Instale el programa de arranque GRUB en /dev/sda.

Ajuste del reloj de sistema en UTC.

Finalice la instalación; el sistema debe reiniciar.

Inicie sesión con la cuenta de usuario bob y cambie a root:

```
Debian GNU/Linux 8 debsrv tty1
debsrv login : bob
password :
bob@debsrv:~$ sudo -i
[sudo] password for bob:
root@debsrv:~#
```


3. Podemos ver el particionado de esta forma:

```
root@debsrv:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	8G	0	disk	
└─sda1	8:1	0	953M	0	part	/boot
└─sda2	8:2	0	1K	0	part	
└─sda5	8:5	0	7,1G	0	part	
└─sysVG-swapLV	254:1	0	976M	0	lvm	[SWAP]
└─sysVG-usrLV	254:2	0	4,7G	0	lvm	/usr
└─sysVG-varLV	254:3	0	4G	0	lvm	/var
sdb	8:16	0	10G	0	disk	
└─sdb1	8:17	0	10G	0	part	
└─sysVG-rootLV	254:0	0	7,5G	0	lvm	/
└─sysVG-varLV	254:3	0	4G	0	lvm	/var
sdc	8:32	0	5G	0	disk	
sdd	8:48	0	10G	0	disk	
sde	8:64	0	9G	0	disk	
sr0	11:0	1	1024M	0	rom	

2. Particionado post-instalación

1. Cree un PV /dev/sdc:

```
#root@debsrv:~# pvcreate /dev/sdc
```

Physical volume "/dev/sdc" successfully created

Verifique:

```
#root@debian:~# pvs
```

PV	VG	Fmt	Attr	PSize	Pfree
/dev/sda5	sysVG	lvm2	a--	7,06g	0
/dev/sdb1	sysVG	lvm2	a--	10,00g	0
/dev/sdc		lvm2	---	5,00g	5,00g

2. Agréguelo en el PV sysVG.

Muestre la configuración del VG. Verá que cuenta con 17 GB.

```
root@debsrv:~# vgs
VG      #PV #LV #SN Attr   VSize  Vfree
sysVG   2   4   0 wz--n- 17,06g  0
```

Agregue el PV a sysVG:

```
root@debsrv:~# vgextend sysVG /dev/sdc
Volume group "sysVG" successfully extended
```

Muestre de nuevo la configuración del VG. Verá que ahora cuenta con 22 GB.

```
root@debsrv:~# vgs
VG      #PV #LV #SN Attr   VSize  Vfree
sysVG   3   4   0 wz--n- 22,05g 5,00g
```

Muestre la configuración de los PV:

```
root@debsrv:~# pvs
PV          VG      Fmt  Attr PSize  Pfree
/dev/sda5   sysVG  lvm2 a--   7,06g   0
/dev/sdb1   sysVG  lvm2 a--  10,00g   0
/dev/sdc    sysVG  lvm2 a--   5,00g 5,00g
```

3. Aumente el tamaño del volumen lógico varLV en 3 GB.

Muestre la configuración de las PV. Podrá constatar que varLV cuenta con 4 GB.

```
root@debsrv:~# lvs
LV      VG      Attr              LSize   Pool Origin Data%  Meta%
Move Log Cpy%Sync Convert
rootLV  sysVG  -wi-ao-----   7,45g
swapLV  sysVG  -wi-ao-----  976,00m
usrLV   sysVG  -wi-ao-----   4,66g
varLV   sysVG  -wi-ao-----   4,00g
```

Aumente el tamaño del LV :

```
root@debsrv:~# lvextend -L+3G /dev/sysVG/varLV -r
Size of logical volume sysVG/varLV changed from 4,00 GiB (1024
extents) to 7,00 GiB (1792 extents).
Logical volume varLV successfully resized
resize2fs 1.42.12 (29-Aug-2014)
El sistema de ficheros de /dev/mapper/sysVG-varLV está montado en /var ;
el cambio de tamaño debe realizarse en línea
old_desc_blocks = 1, new_desc_blocks = 1
El sistema de ficheros en /dev/mapper/sysVG-varLV tiene ahora un tamaño
de 1835008 bloques (4k).
```

Muestre la configuración de LV, se aprecia que varLV tiene ahora 7 GB.

```
root@debsrv:~# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
rootLV  sysVG  -wi-ao---- 7,45g
swapLV  sysVG  -wi-ao---- 976,00m
usrLV   sysVG  -wi-ao---- 4,66g
varLV   sysVG  -wi-ao---- 7,00g
```

4. Cree dos PV /dev/sdd y /dev/sde:

```
root@debsrv:~# pvcreate /dev/sdd /dev/sde
Physical volume "/dev/sdd" successfully created
Physical volume "/dev/sde" successfully created
```

5. Cree el VG appVG y añada los dos PV:

```
root@debsrv:~# vgcreate appVG /dev/sdd /dev/sde
Volume group "appVG" successfully created
```

6. Cree dos LV: progLV y dataLV.

Contamos con 18,99 GB para crear dos LV:

```
root@debsrv:~# vgs
VG      #PV #LV #SN Attr   VSize  Vfree
appVG   2   0   0 wz--n- 18,99g 18,99g
sysVG   3   4   0 wz--n- 22,05g 2,00g
```

Tomamos 5 GB para progLV y 8 GB para dataLV, por ejemplo:

```
root@debsrv:~# lvcreate -n progLV -L5G appVG
root@debsrv:~# lvcreate -n dataLV -L8G appVG
```

7. Formatee progLV en ext4 y dataLV en ext3.

```
root@debsrv:~# mkfs.ext4 /dev/appVG/progLV
root@debsrv:~# mkfs.ext3 /dev/appVG/dataLV
```

8. Monte los dos LV automáticamente al arrancar el servidor.

Cree dos directorios en /mnt para los dos puntos de montaje:

```
root@debsrv:~# mkdir /mnt/prog /mnt/data
```

Modifique el archivo /etc/fstab y añada las dos líneas siguientes:

```
/dev/mapper/appVG-progLV /mnt/prog      ext4    defaults    0    2
/dev/mapper/appVG-dataLV /mnt/data    ext3    defaults    0    2
```

Reinicie e inicie de nuevo una sesión con la cuenta bob. Cambie de identidad a root. Compruebe el montaje de los dos LV:

```
root@debsrv:~# mount | grep appVG
/dev/mapper/appVG-progLV on /mnt/prog type ext4
(rw,relatime,data=ordered)
/dev/mapper/appVG-dataLV on /mnt/data type ext3
(rw,relatime,data=ordered)
```

9. Cree el directorio infos en dataLV:

```
root@debsrv:~# mkdir /mnt/data/infos
```

10. Cree cuatro archivos arch01, arch02, arch03 y arch04 en el directorio infos:

```
root@debsrv:~# touch /mnt/data/infos/arch0{1,2,3,4}
```

11. Convierta dataLV en ext4 sin perder los datos:

```
root@debsrv:~# tune2fs -O extents,uninit_bg,dir_index  
/dev/appVG/dataLV
```

Corrija el archivo /etc/fstab. Reemplace ext3 por ext4:

```
/dev/mapper/appVG-dataLV /mnt/data      ext4      defaults    0    2
```

Si escribe el comando mount, observará que el montaje sigue siendo ext3.

```
root@debsrv:~# mount |grep dataLV  
/dev/mapper/appVG-dataLV on /mnt/data type ext3  
(rw,relatime,data=ordered)
```

Desmonte y vuelva a montar dataLV:

```
root@debsrv:~# umount /dev/appVG/dataLV  
root@debsrv:~# mount /dev/appVG/dataLV  
/mnt/data/  
root@debsrv:~# mount |grep dataLV  
/dev/mapper/appVG-dataLV on /mnt/data type ext4  
(rw,relatime,data=ordered)
```

Ejercicio del capítulo Secuencia de arranque

Debemos utilizar la máquina virtual CentOS 7.

1. Modifique el retardo por defecto del menú GRUB de forma permanente.

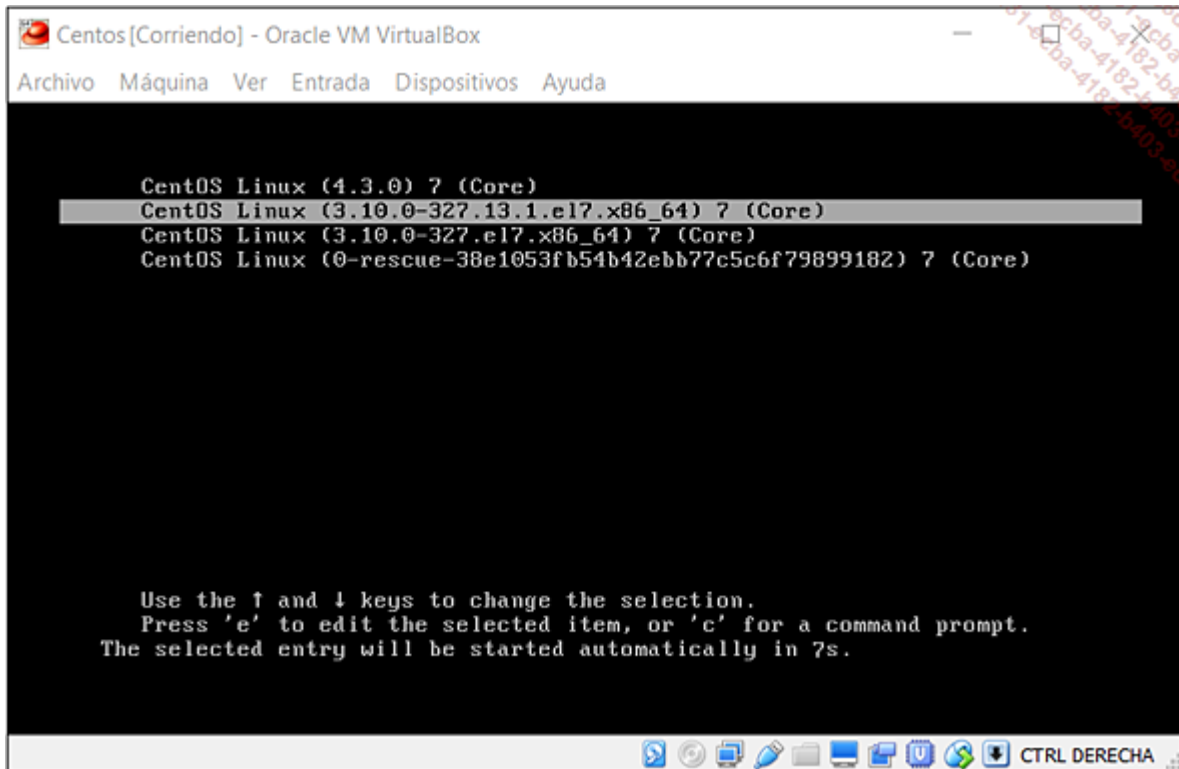
Edite el archivo /etc/default/grub y cambie el retardo a 8 segundos:

```
GRUB_TIMEOUT=8
```

Regenere la configuración de GRUB:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

2. Reinicie el sistema para verificar:



Ejercicio del capítulo

Mantenimiento de la configuración de red

1. Recopilar la configuración de una tarjeta de red

Se necesitan dos máquinas virtuales:

- la primera es una máquina Debian o Ubuntu Server.
- la segunda es una máquina CentOS.

1. Utilice la primera. Inicie una sesión como usuario root.

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password : bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

2. Identifique los siguientes datos:

Elemento	Valor	Comando utilizado
Nombre de la tarjeta	eth0	lshw -c network
Nombre del fabricante y modelo	82540EM Gigabit Ethernet Controller Intel Corporation	lshw -c network
Controlador de la tarjeta	e1000	lshw -c network
Dúplex	full	lshw -c network
Velocidad de la tarjeta	1000	lshw -c network

Dirección MAC	08:00:27:3a:17:d8	Ip addr
Dirección estática o dinámica		/etc/network/interfaces
Dirección IP y máscara	10.0.2.15/24	Ip addr
Gateway por defecto	10.0.2.2	Netstat -rn
Nombre del equipo	debsrv	hostname

Otros comandos también pueden proporcionar respuestas adecuadas.

2. Renombrar la interfaz de red eth0

1. Utilice la segunda máquina virtual.

Inicio de sesión:

```
CentOS Linux 7 (core)
Kernel 3.10.0-229.20.1.el7.x86_64 on an X86_64

srv02 login : root
Password :
[root@srv02 ~]#
```

2. Identifique el nombre de la tarjeta de red cableada. Luego cambie el nombre a eth0. Describa su procedimiento.

El nombre actual de la tarjeta de red:

```
# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.16 netmask 255.255.255.0 broadcast
192.168.1.255
    inet6 fe80::a00:27ff:fe8d:fdb prefixlen 64 scopeid
0x20<link>
    ether 08:00:27:8d:0f:db txqueuelen 1000 (Ethernet)
    RX packets 131 bytes 16837 (16.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
```



```
TX packets 145  bytes 18999 (18.5 KiB)
```

```
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

```
inet 127.0.0.1  netmask 255.0.0.0
```

```
inet6 ::1  prefixlen 128  scopeid 0x10<host>
```

```
loop txqueuelen 0  (Bucle locale)
```

```
RX packets 147  bytes 13196 (12.8 KiB)
```

```
RX errors 0  dropped 0  overruns 0  frame 0
```

```
TX packets 147  bytes 13196 (12.8 KiB)
```

```
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

El nombre de la interfaz es enp0s3.

Para cambiar el nombre de esta interfaz, debe abrir el archivo `/etc/default/grub` con un editor de texto.

Busque la línea que empieza por `GRUB_CMDLINE_LINUX` y agregue al final de esta línea `net.ifnames=0`. Luego, guarde el archivo.

Vuelva a generar la configuración de GRUB escribiendo:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

En el directorio `/etc/sysconfig/network-scripts/`, hay que renombrar el archivo de configuración de la interfaz:

```
# mv ifcfg-enp0s3 ifcfg-eth0
```

Modifique con `sed` el valor del campo `name` en cada archivo:

```
# sed -i 's/enp0s3/eth0/g' ifcfg-eth0
```

Ejercicio del capítulo Análisis de rendimiento

Vamos a utilizar la máquina virtual Debian.

Inicie una sesión en línea de comandos con el usuario root.

Para la distribución Debian:

```
Debian GNU/Linux 8 srv01 tty1
srv01 login : bob
password :
bob@srv01:~$ sudo -i
[sudo] password for bob:
root@srv01:~#
```

1. Creación de un archivo swap

1. Observe el estado actual del swap y apunte los valores.

```
# free -mt
```

	total	used	free	shared	buffers	cached
Mem:	490	124	365	4	12	50
-/+ buffers/cache:		61	428			
Swap:	975	0	975			
Total:	1466	124	1341			

2. Cree un archivo swap de 100 MB llamado mem.swp.

```
# dd if=/dev/zero of=mem.swp bs=1024 count=100000
```

3. Active el archivo swap. Verifique que esté activo.

Cree una zona de swap en el archivo mem.swp:

```
# mkswap mem.swp
```

Asigne permisos adaptados para este archivo:

```
# chmod 600 mem.swp
```

Activación:

```
# swapon -v mem.swp
```

Verificación:

```
# free -mt
```

4. Elimine el archivo swap. Verifique que está eliminado.

Desactive el archivo swap:

```
# swapoff -v mem.swp
```

Luego elimínelo:

```
# rm mem.swp
```

Vuelva a consultar el estado actual del swap:

```
# free -mt | grep Swap
```

2. Prueba de carga de memoria

1. Instale el software Stress desde los repositorios de Debian.

```
# apt-get install stress
```

2. Ejecute el comando stress de la forma siguiente:

```
# stress --vm-bytes $(awk '/MemFree/{printf "%d\n", $2 * 2;}'  
< /proc/meminfo)k --vm-keep -m 1 &
```

Dispone del comando stress que simula una carga de memoria. Funciona en segundo plano.

```
# jobs -l
```

3. Muestre con el comando free el estado de la memoria.

```
# free -mt
```

	total	used	free	shared	buffers	cached
Mem:	490	485	4	0	0	3
-/+ buffers/cache:		481	9			
Swap:	975	475	500			
Total:	1466	961	504			

4. Muestre las estadísticas con vmstat.

```
# vmstat -s
```

501868 K total memory
494216 K used memory
227992 K active memory
227864 K inactive memory
7652 K free memory
904 K buffer memory
3920 K swap cache
999420 K total swap
592104 K used swap
407316 K free swap
41 non-nice user cpu ticks
0 nice user cpu ticks
453 system cpu ticks
116893 idle cpu ticks
11803 IO-wait cpu ticks
0 IRQ cpu ticks
161 softirq cpu ticks
0 stolen cpu ticks
2497453 pages paged in
3109628 pages paged out
585947 pages swapped in
724226 pages swapped out

```
159786 interrupts
402781 CPU context switches
1448819516 boot time
1968 forks
```

5. Recopile la paginación con el comando sar cinco veces cada 10 segundos.

```
# sar -B 10 5
```

```
[root@localhost ~]# sar -B 10 5
Linux 3.10.0-327.13.1.el7.x86_64 (localhost.localdomain)      09/05/16      _x86_64_ (2 CPU)

16:17:28      pgpgin/s pgpgout/s   fault/s majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s   %vmeff
16:17:38          0,00    23,80      8,40      0,00  1217,20      0,00      0,00      0,00      0,00
16:17:48          2,40      0,40    198,80      0,00  4990,00      0,00      0,00      0,00      0,00
16:17:58          0,00      1,20      9,90      0,00  1487,70      0,00      0,00      0,00      0,00
16:18:08          0,00      2,20      3,30      0,00  1575,10      0,00      0,00      0,00      0,00
16:18:18         20,00      0,80    281,00      0,00  1460,30      0,00      0,00      0,00      0,00
Media:          4,48      5,68    100,28      0,00  2146,06      0,00      0,00      0,00      0,00
```

6. Mate el proceso stress ejecutado en segundo plano. Verifique.

```
# pkill --signal 9 stress
# jobs -l
```

7. Muestre con el comando free el estado de la memoria.

```
# free -mt
```

	total	used	free	shared	buffers	cached
Mem:	490	105	384	0	0	6
-/+ buffers/cache:		98	391			
Swap:	975	106	869			
Total:	1466	212	1253			