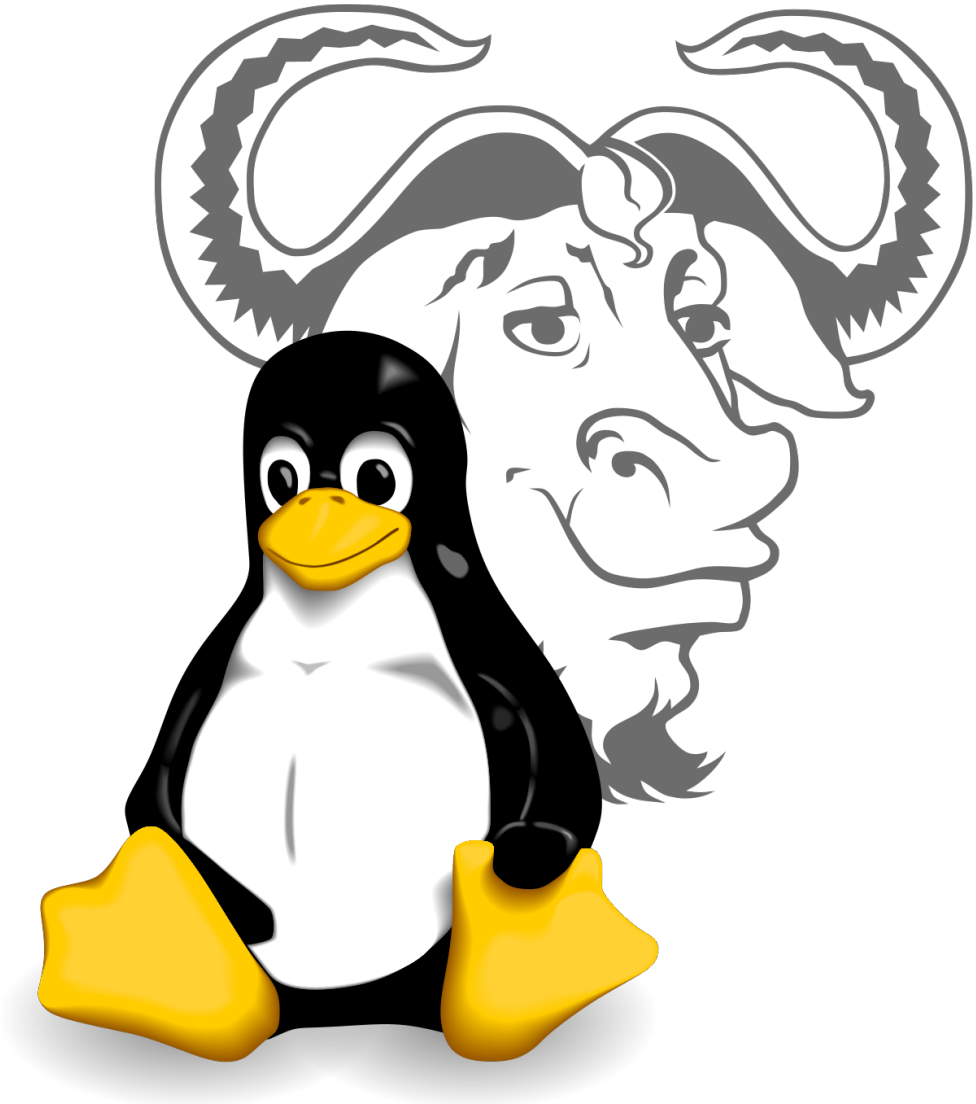
Universidad de Salamanca

Grado en Ingeniería Informática

Administración de Sistemas



Administración de Sistemas GNU Linux

Autor

Mr Hydden

Ficheros y sistema de ficheros

1 Vista general del sistema de ficheros de Linux

1.1 Ficheros

1.1.1 General

«**En un sistema UNIX, todo es un fichero; si algo no es un fichero, es un proceso**». Esta es una generalización aceptable. En UNIX hay ficheros especiales (e.g., tuberías con nombre –named pipes –, sockets, etc.), pero no hay diferencias entre directorios y ficheros (aquéllos son ficheros que contienen nombres de otros ficheros; los dispositivos de E/S también son ficheros desde el punto de vista del sistema.

Con vistas a entender el sistema de ficheros, es bueno pensarlo como una **estructura arborescente** en el disco duro; no obstante, posteriormente veremos que esta imagen no es completamente exacta.

1.1.2 Tipos de ficheros

La mayoría de los ficheros contiene datos, como texto, programas, etc. Estos se denominan **ficheros normales** (*regular files*). Otros tipos de ficheros que encontramos en el sistema son:

- **Directorios**: ficheros que contienen una lista de nombres de otros ficheros
- **Ficheros especiales**: usados para E/S; casi todos se encuentran en /dev. Hablaremos de ellos posteriormente.
- **Enlaces (*links*)**: permiten acceder a un fichero o directorio desde diferentes lugares del sistema. Trataremos el tema más adelante.
- **Sockets (de UNIX)**: similares a sockets TCP/IP, proporcionan comunicación entre procesos protegida por el control de acceso del sistema de ficheros.
- **Tuberías con nombre (*named pipes*)**: parecidos a los sockets; permiten comunicar procesos entre sí sin utilizar la semántica de los sockets.

Es posible ver el tipo de un fichero en el sistema mediante la opción -l de la orden ls. Los tipos de ficheros se muestran, en el primer símbolo de la primera columna de la salida de ls como:

```
(mrhydden)-(0)-(10:16:08)
(~)-$ ls -l
total 4
-rw-r--r-- 1 mrhydden mrhydden    0 feb 19 11:49 file
drwxr-xr-x 2 mrhydden mrhydden 4096 feb 19 11:49 folder
lrwxrwxrwx 1 mrhydden mrhydden    4 feb 19 11:49 link -> file
```

Símbolo	Tipo de fichero
-	Fichero normal
d	Directorio
l	Enlace
c	Fichero especial
s	Socket
p	Tubería con nombre
b	Dispositivo de bloques

SUGERENCIA: echar un vistazo a las opciones -F y --color de la orden ls.

1.2 Particiones

1.2.1 ¿Por qué particionar?

Una partición es el nombre que recibe cada **división lógica de una única unidad física de almacenamiento de datos** (e.g., un disco duro). **Una partición tiene un formato**, es decir, un modo de organización interna, que depende del sistema de ficheros utilizado. En general, un sistema operativo interpreta, utiliza y manipula cada partición de un dispositivo físico como un dispositivo independiente.

Entre los principales objetivos de dividir un dispositivo en diferentes particiones está el de **proporcionar seguridad** en caso de accidente; de este modo, si hay algún problema, sólo la partición afectada resultará dañada.

Este principio data de los tiempos en los que Linux aún no contaba con un sistema de ficheros con registro[1] (*journaling; journaled file system*). A día de hoy se sigue utilizando fundamentalmente para aportar robustez adicional. Nótese que el *journaling* sólo proporciona seguridad frente a fallo por desconexión; los errores lógicos en los bloques sólo se pueden detectar y corregir mediante el uso de RAID.

1.2.2 Organización y tipos de particiones

Hay dos tipos principales de particiones en un sistema Linux:

- **Partición de datos**: contiene datos normales del sistema.
- **Partición de intercambio (*swap*)**: extensión de la memoria principal del computador, utilizada por el subsistema de memoria virtual.

La mayoría de los sistemas tienen una partición *root*, una o más particiones de datos y una o más particiones *swap*.

En general, los sistemas Linux usan la utilidad fdisk en el momento de su instalación para gestionar las particiones.

1 El *journaling* es un sistema de registro de transacciones, que almacena la información necesaria para restablecer los datos de la transacción en caso de fallo. Se emplea por extenso en sistemas operativos, sistemas gestores de bases de datos, etc.

Una partición *root* estándar tiene en torno a 100 – 500 MB, y contiene ficheros de configuración del sistema, órdenes y otros programas básicos, bibliotecas del sistema, cierto espacio temporal y el directorio *home* del usuario administrador.

El espacio *swap* es accesible únicamente para el sistema, y no es visible durante la operación normal. En general, el sistema supone que tendrá aproximadamente el doble de espacio en *swap*de su tamaño de memoria principal.

También es posible para algunas distribuciones de Linux que aparezca una partición /boot, que contiene el kernel y algunos ficheros anexos a él.

El resto del almacenamiento se suele dividir en particiones de datos. Un esquema habitual es tener particiones para /usr, /home, /var y /opt.

En general es **buena idea separar los datos de usuario y los de programas**. No obstante en equipos monousuario hay más variabilidad a este respecto.

1.2.3 Puntos de montaje

Un **punto de montaje** es un **directorio que define el lugar en que un conjunto de datos se encuentra en el sistema**; todas las particiones se encuentran asociadas a un punto de montaje. Habitualmente todas las particiones se conectan a través de la partición *root* (/).

Para asociar una partición a un punto de montaje se utiliza la orden mount. Para la operación inversa se dispone de la orden umount.

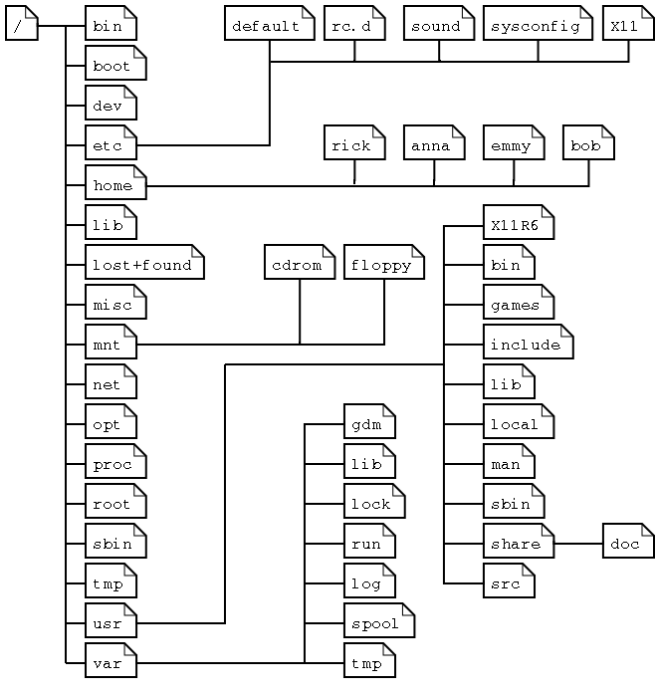
Durante el arranque del sistema todas las particiones se montan así, tal y como se describe en el fichero de configuración /etc/fstab.

En un sistema en funcionamiento se puede ver información sobre las particiones y sus puntos de montaje mediante la orden df.

1.3 Organización del sistema de ficheros

1.3.1 Visión global

El «Estándar de jerarquía del sistema de ficheros» (*Filesystem Hierarchy Standard, FHS*) define una estructura de directorios en las distribuciones de Linux del modo siguiente:



Dependiendo de la distribución y el administrador de sistema esto puede variar. Especialmente importantes son los directorios contenidos en el directorio *root* (/). Entre los más habituales encontramos:

Directorio	**Contenido**
/bin	Programas habituales, compartidos entre el sistema, los usuarios y el administrador.
/boot	Los ficheros de *startup* y el núcleo, vmlinuz. En algunas distribuciones más nuevas también hay datos de grub.
/dev	Contiene referencias a los periféricos, que el sistema representa como ficheros con propiedades especiales.
/etc	Contiene los ficheros principales de configuración del sistema.
/home	Directorios de usuarios normales.
/initrd	(sólo en algunas distribuciones) Información de boot.
/lib	Ficheros de bibliotecas, necesarios para todo tipo de programas y usuarios.
/lost+found	Existe una para cada partición. Los ficheros que fueron guardados durante un error se encuentran aquí.
/misc	Para propósitos variados.
/mnt	Punto de montaje estándar para sistemas de ficheros externos (e.g., una memoria USB o una cámara).
/net	Punto de montaje estándar para sistemas de ficheros remotos completos.
/opt	Típicamente, contiene software adicional y de terceros.
/proc	Sistema de ficheros virtual que contiene información sobre los recursos

	del sistema. El fichero proc.txt explica el sistema de ficheros virtual por extenso.
/root	El directorio *home* del usuario administrador.
/sbin	Programas que sólo son usados por el sistema y el administrador del mismo.
/tmp	Espacio temporal que utiliza el sistema. Se borra con cada reinicio.
/usr	Siglas de «Unix System Resources» ('Recursos del sistema Unix). Contiene programas, bibliotecas, documentación, etc.
/var	Almacena ficheros variables y temporales creados por los usuarios, tales como ficheros log, colas de correo electrónico, zona de spooling, ficheros descargados de Internet, etc.

SUGERENCIA: echar un vistazo a **man hier**

1.3.2 El sistema de ficheros real

Aunque resulta útil pensar el sistema de ficheros como una estructura arborescente, el sistema no lo entiende como tal:

Cada partición tiene su propio sistema de ficheros. En un sistema de ficheros, un fichero es representado por un ***inodo***, que es una especie de número de serie con meta-información del fichero (localización en el disco, propiertario, etc.). Cuando un disco duro se formatea, se le asocia un número fijo de *inodos* por partición, que será el máximo número de ficheros que pueden existir en la partición; típicamente, cada *inodo* ocupa 2 – 8 kB. Cuando un fichero se crea *de novo*, se le asocia un *inodo* libre.

La POSIX establece que un *inodo* tiene un UID en el sistema de ficheros, que junto al UID del dispositivo, identifica de modo único al *inodo* en el sistema. La **estructura de un *inodo*** según ese estándar es:

- Identificador de dispositivo
- Identificador de fichero
- «Modo» del fichero (información sobre propietarios y permisos de acceso)
- Cuenta de enlaces (indica cuántos enlaces duros – ver abajo – existen)
- Identificador del propietario y de su grupo
- Identificador del dispositivo (si el fichero es un fichero especial de dispositivo)
- Tamaño del fichero
- Varios *timestamps* (cuándo fue modificado el *inodo* – *ctime* –, cuándo fue modificado el contenido del fichero – *mtime* –, cuándo se accedió a él por última vez – *atime* –).
- El tamaño de bloque de E/S preferido
- El número de bloques reservados para el fichero

Nota bene: nótese que el nombre del fichero y el directorio no se almacenan aquí. El único lugar en el que aparece esa información es en los ficheros especiales directorio.

SUGERENCIA: echar un vistazo a las órdenes **ls -i** *y* **stat**

2 Orientarse en el sistema de ficheros

2.1 Path

La variable de entorno PATH contiene una lista de directorios en los que es posible encontrar ficheros ejecutables. Es por eso que es posible ejecutar programas sin tener que escribir la ruta completa hasta ellos (e.g., el programa ls se encuentra en /bin/ls).

```
(mrhydden)-(0)-(10:16:08)
(~)-$ echo $PATH
/etc/bin/anaconda2/bin:/home/mrhydden/bin:/home/mrhydden/.local/bin:/usr/
local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/snap/bin
```

2.2 Ficheros y directorios principales

2.2.1 El núcleo del sistema operativo

El fichero /vmlinux es un fichero ejecutable enlazado estáticamente que contiene el núcleo del sistema operativo en un formato ejecutable soportado por Linux (ELF, COFF, o a.out). También es posible encontrarlo en /boot, debido a que algunos sistemas i386 están limitados a poder direccionar los primeros 1024 cilindros desde el BIOS, por lo que es necesaria una partición especial para /boot al principio del disco para que sea posible acceder a él.

2.2.2 La shell

Es difícil dar una definición clara de «*shell*». Aún así, es una buena aproximación decir que la *shell* es una manera de comunicarse con el ordenador. La mayoría de los usuarios conocen otro lenguaje: el de señalar-y-hacer-click. Sin embargo, ese es un lenguaje mucho más rígido, y una GUI nunca puede alcanzar las capacidades de las órdenes de la *shell*, que forma el backend. Con una definición un poco más amplia, podemos llamar *shell* a la interfaz entre el usuario y el sistema; así una *shell* sería una «concha» que recubre la complejidad del sistema, y como tal podemos tener al menos tres tipos: *Command-Line Interface*, *Graphical User Interface* y *Natural User Interface* (la primera definición que dábamos corresponde más bien con una *shell* CLI).

Hay diferentes *programas shell*:

- sh (Bourne Shell): la *shell* original de UNIX. Es básica y pequeña. Puede emularse con bash (cf. infra) en modo compatible con POSIX.
- bash (Bourne Again Shell): la *shell* estándar de GNU, intuitiva y flexible. Es la más utilizada. Es una extensión de sh (todo lo que funciona en sh, funciona en bash). Los ejemplos de estos apuntes utilizan bash.
- csh (C Shell): una *shell* cuya sintaxis es parecida al lenguaje C.
- tcsh (Turbo C Shell): una extensión de csh, más amigable y potente.
- ksh (Korn Shell): otra extensión de sh.

El fichero /etc/shells contiene una lista de las *shells* reconocidas por el sistema. La shell por defecto de un usuario se almacena en el fichero /etc/passwd.

Para cambiar de shell, basta por escribir el nombre de la *shell* que se desea en la *shell* actual:

```
(mrhydden)-(0)-(10:16:08)
(~)-$ tsh
[mrhydden@the-void ~]$
```

La variable de entorno $SHELL almacena la *shell* en uso.

NOTA: habitualmente /bin/sh es una falsa sh; en realidad es un enlace a bash, que se ejecuta en modo compatible con POSIX.

2.2.3 El directorio 'home'

Este es el directorio por defecto de entrada al sistema. En general, es un subdirectorio de /home (aunque puede estar en otro lugar, por ejemplo en un servidor remoto, etc.). En cualquier caso, la ruta al directorio *home* está almacenada en la variable de entorno $HOME. En la especificación de rutas, la ruta absoluta a *home* se puede representar mediante una virgulilla (~), por ejemplo: ~/Documents/somefile.txt.

Aunque se puede utilizar libremente el directorio *home*, su tamaño puede estar limitado. Ese límite se puede ver mediante la orden quota (puede no estar limitado).

2.3 Ficheros de configuración principales

Como dijimos, la mayoría de los ficheros de configuración se encuentra en el directorio /etc. Entre los más importantes están los que indicamos a continuación; no obstante, se puede ver más información accediendo a ellos mediante la orden cat; se podrá apreciar si se hace esto que en general son bastante claros y bien documentados.

Fichero	Información/Servicios
aliases	Alias para el uso con los servidores de correo electrónico Sendmail o Postfix; mapean nombres de usuario a direcciones de correo electrónico.
apache	Ficheros de configuración del servidor web Apache.
bashrc	Fichero de configuración – para todo el sistema – de la Bourne Again Shell. Define funciones y alias para todos los usuarios. Otras *shell* pueden tener sus propios ficheros (e.g., cshrc).
crontab y directorios cron.*	Configuración de tareas que deben ejecutarse periódicamente – copias de seguridad, actualizaciones, limpieza del sistema, etc. –
default	Opciones por defecto de algunas órdenes, por ejemplo useradd.

filesystems	Sistemas de ficheros conocidos: ext3, vfat, iso9660, etc.
fstab	Lista de particiones y sus puntos de montaje.
ftp*	Configuración del servidor ftp (quién puede conectarse, que partes del sistema son accesibles, etc.)
group	Fichero de configuración de grupos de usuarios. Órdenes como groupadd, groupmod o groupdel editan este fichero.
hosts	Lista de máquinas que pueden encontrarse utilizando la red pero sin necesidad de DNS. *No confundir con la configuración de red del sistema, que está en* /etc/sysconfig
inittab	Información de arranque: modo, número de consolas de texto, etc.
issue	Información sobre la distribución (versión y/o información sobre el núcleo).
ld.so.conf	Localización de ficheros de bibliotecas.
lilo.conf, silo.conf, aboot.conf, etc.	Información de aranque del Linux Loader. Actualmente se pueden encontrar datos de GRUB también.
logrotate.*	Configuración de la rotación de logs[2]
mail	Directorio que contiene instrucciones sobre el funcionamiento del servidor de correo electrónico.
modules.conf	Configuración de módulos que habilitan características especiales (*drivers*).
motd	Message Of The Day: se muestra a todo el mundo que se conecta al sistema en modo texto. Puede usarse para comunicar operaciones de mantenimiento del sistema, etc.
mtab	Información sobre los sistemas de ficheros montados actualmente.
nsswitch.conf	Información sobre el acceso ordenado a DNS para resolver un nombre.
pam.d	Configuración de módulos de autentificación.
passwd	Lista de usuarios locales. Herramientas como useradd, usermod y userdel editan este fichero.

2 Mecanismo que evita la acumulación de grandes cantidades de ficheros log. Más información en man logrotate

printcap	Fichero anticuado, pero aún usado para la configuración de impresoras.
profile	Configuración global del entorno de la shell: variables, propiedades por defecto de los ficheros nuevos, limitación de recursos, etc.
rc*	Directorios que definen los servicios activos para cada nivel de ejecución.
resolv.conf	Configuración del programa cliente (*resolver*) de DNS.
sendmail.cf	Fichero principal de configuración del servidor de correo Sendmail.
services	Conexiones aceptadas por el sistema (puertos abiertos).
sndconfig o sound	Configuración de la(s) tarjeta(s) de sonido y los eventos de sonido.
ssh	Directorio de configuración del cliente y el servidor SSH.
sysconfig	Directorio que contiene ficheros de configuración del sistema: ratón, teclado, redes, escritorio, gestión de la energía, etc.
X11	Configuración del servidor gráfico X Window. También contiene indicaciones generales para los gestores de ventanas disponibles: gmd, fvwm, twm, etc.
xinetd.* o inetd.conf	Ficheros de configuración de servicios de Internet que se ejecutan en el *daemon* de servicios de Internet (que no se ejecutan por tanto como *daemons* independientes).

2.4 Dispositivos más comunes

Los dispositivos (i.e., cada periférico que se asocia al sistema y es diferente de la CPU) se presenta como una entrada en el directorio /dev. En general, el sistema operativo gestiona toda la información, de modo que libera al usuario de la preocupación de llevar a cabo esas tareas. Indicamos aquí una lista de dispositivos habituales:

Nombre	**Dispositivo asociado**
cdrom	Disco compacto
console	Entrada especial para la consola en uso
cua*	Puertos serie
dsp*	Dispositivos de muestreo y grabación

fd*	Entradas para disquetes
hd[a-t][1-16]	Soporte estándar para almacenamiento IDE
ir*	Dispositivos infrarrojos
isdn*	Gestión de conexiones ISDN
js*	Joysticks
lp*	Impresoras
mem	Memoria
midi*	Reproductor midi
mixer* and music	Modelo de un mezclador
modem	Módem
mouse	Ratones
null	Pseudo-dispositivo[3] que acepta todo lo que entra en él
par*	Soporte de puertos paralelos
pty*	Pseudo terminales
radio*	Para dispositivos de radiofrecuencia amateur
ram*	Dispositivo de arranque (*boot*)
sd*	Discos SCSI con sus particiones
tty*	Consolas virtuales (emuladores de VT100)
usb*	Tarjetas usb
video*	Para tarjetas gráficas

2.5 Ficheros variables

En el directorio /var se almacenan datos variables de tipos específicos (ficheros de log, bandejas de correo electrónico, spoolers, etc.).

Dado que habrá mucha actividad en estos ficheros (e incluso parte será generada por usuarios anónimos de Internet) es frecuente crear una partición aparte para /var.

Anexo I: órdenes de shell y su uso

3 Los dispositivos en sistemas tipo UNIX no tienen por qué corresponderse con dispositivos físicos. Los ficheros que no tienen un dispositivo físico asociado se denominan pseudo-dispositivos (*pseudo-devices*). Estos proporcionan diferente funcionalidad que gestiona el sistema operativo. Entre los más usados están:

- /dev/null: acepta y descarta todos los datos de entrada, y devuelve EOF si se intenta leer de él.
- /dev/zero: acepta y descarta todos los datos de entrada, y produce un flujo continuo de caracteres nulos (NUL) si se lee de él
- /dev/full: produce un continuo flujo de caracteres nulos si se lee de él y devuelve «disk full» si se intenta escribir
- /dev/random: produce un flujo continuo de números aleatorios de tamaño variable cuando se lee de él

1 Gestión de paquetes[4]

1.1 dpkg

dpkg es el programa principal de gestión de paquetes Debian. dpkg permite inspeccionar, instalar, manipular, etc., paquetes binarios .deb. Sin embargo, dpkg simplemente gestiona los paquetes instalados; herramientas más avanzadas como apt gestionan además listas de dependencias que permiten instalar todo lo que sea necesario.

dpkg se utiliza principalmente para instalar paquetes ya disponibles. La instalación se lleva a cabo en dos pasos: desempaquetado y configuración.

```
~$ dpkg -i <binary_pkg_name>.deb
```

A veces dpkg emitirá un error. Es posible ignorarlo mediante las opciones --force-*.

Es posible eliminar un paquete con la opción -r. Sin embargo, la eliminación no es completa: se conservan archivos de configuración, scripts, registros de sistema y datos de otros usuarios. Así es posible reinstalarlo rápidamente con la misma configuración. Si se desea eliminar toda esta información se debe utilizar la opción -P en lugar de -r.

dpkg gestiona una base de datos de los paquetes del sistema. Para gestionarla se dispone de las siguientes opciones:

- -L | --listfiles <package_name>: muestra la lista de ficheros instalados por el paquete.
- -S | --search <filename>: busca los paquetes que contienen el fichero con el nombre indicado.
- -l | --list: muestra la lista de paquetes conocidos por el sistema y su estado de instalación.
- -s | --status <package_name>: muestra las cabeceras de un paquete instalado.
- -c | --contents <binary_pkg_name>.deb: lista los archivos contenidos en el paquete Debian especificado.
- -I | --info <binary_pkg_name>.deb: muestra las cabeceras del paquete Debian especificado.

Por último, resulta relevante señalar que dpkg mantiene un registro de todas sus acciones en /var/log/dpkg.log. Esto permite, entre otras cosas, mantener un historial del desarrollo del sistema, siendo posible localizar el momento en que se instaló o actualizó un paquete, etc.

Anexo II: Master Boot Record (MBR)

Se llama *sector de arranque* (*boot sector*) a una región del disco duro, disco óptico, o cualquier otro dispositivo de almacenamiento que contiene código máquina para permitir al proceso de arranque de un sistema cargar un programa (habitualmente un sistema operativo) almacenado en el dispositivo.

En los computadores compatibles con el IBM PC (i.e., la mayoría de los ordenadores personales) existen dos tipos de sector de arranque principales:

4 La lectura de esta sección resultará más provechosa a aquellos lectores que tengan conocimientos de la estructura de un paquete binario de Debian.

- El *Volume Boot Record* (VBR) que es el primer sector en un dispositivo que no ha sido particionado, o el primer sector de una partición en un dispositivo particionado. Puede contener código para cargar el sistema operativo instalado en el dispositivo o partición.
- El *Master Boot Record* (MBR) que es el primer sector de un dispositivo particionado y permite localizar la partición activa y invocar su VBR.

El soporte para medios con particiones (i.e., el MBR) fue introducido con el IBM PC DOS 2.0 en 1983, junto con la herramienta FDISK (de DOS) para configurar y gestionar las particiones MBR.

MBR y arranque del sistema en computadores compatibles con el IBM PC

En este tipo de sistemas, el BIOS ignora diferencias entre MBR y VBR. El firmware de arranque (*bootstrapping*) simplemente se encarga de cargar el MBR en memoria principal e indicar a la CPU que comience a ejecutarlo (de ahí que numerosos ataques maliciosos intenten acceder al sector de arranque). La única condición necesaria es que exista la firma 0x55AA en los últimos dos bytes del sector. El código en el MBR sí que entiende de particiones, y se encarga de cargar y ejecutar el VBR de la partición activa, el cual a su vez suele cargar un *bootloader* diferente. En general el MBR se carga en la dirección de memoria 0x0...07C00.

Dado que los discos tradicionalmente tienen sectores de 512 bytes y el MBR ocupaba un sector, el número de particiones se limitó a cuatro. La estructura de un MBR típico es la siguiente:

<table>
<tr><th>Offset</th><th colspan="2">Contenido</th><th>Tamaño (bytes)</th></tr>
<tr><td>0x0</td><td colspan="2">Área de código de arranque (boot loader)</td><td>446</td></tr>
<tr><td>0x1BE</td><td>Tabla de partición #1</td><td rowspan="4">Tablas de particiones de las particiones primarias</td><td>16</td></tr>
<tr><td>0x1CE</td><td>Tabla de partición #2</td><td>16</td></tr>
<tr><td>0x1DE</td><td>Tabla de partición #3</td><td>16</td></tr>
<tr><td>0x1EE</td><td>Tabla de partición #4</td><td>16</td></tr>
<tr><td>0x1FE</td><td>0xAA55</td><td>Firma (boot signature)</td><td>2</td></tr>
</table>

A su vez, las tablas de particiones tienen el formato siguiente:

Tamaño (bytes)	Contenido
1	Estado de la partición: habitualmente 0x80, código para “activa” (*bootable*)
3	Inicio de la partición, dirección CHS: terna (cilindro, cabezal, sector)
1	Tipo de partición (00 – inválida, 01 – FAT12, 02 – XENIX, 05 – extendida, 06 – FAT16, etc.)
3	Fin de la partición dirección CHS
4	LBA (*Logical Block Addressing*) del primer sector

4	Número de sectores en la partición

UEFI y GPT[5]: más allá del MBR

Debido a las limitaciones de las tablas de particiones del MBR, con el tiempo se llevó a cabo una transición a GPT (*GUID Partition Table*), otro estándar de definicón de tablas de particiones mediante identificadores globalmente únicos (*GUID*), que forma parte del estándar UEFI (||*Unified Extensible Firmware Interface*, desarrollado como reemplazamiento del BIOS del IBM PC). Todos los ordenadores personales modernos soportan GPT.

Las principales limitaciones del MBR que comentábamos anteriormente son:

- Limitación a cuatro particiones (por el límite de un MBR de 512 bytes y tablas de 16 bytes)
- Tamaño máximo direccionable de 2TiB
- Máximo número de tipos de partición: 256 tipos
- Direccionamiento CHS es muy poco flexible
- No hay seguridad en las tablas (backups, checksums, etc.)

GPT solventa todos estos problemas, proporcionando las siguientes características:

- Número variable de particiones
- Tamaño máximo direccionable 8ZiB
- Direccionamiento lineal con LBA
- GUID y tipos de particiones ilimitados a efectos prácticos (128 bits)
- Copia de seguridad de la GPT
- MBR de protección frente a configuración antigua (retrocompatible).
- Checksums

5 Para más información sobre UEFI y GPT se puede consultar: https://en.wikipedia.org/wiki/GUID_Partition_Table