

5. Montaje y desmontaje de sistemas de ficheros

El sistema de ficheros se visualiza mejor como un árbol, tal y como está, en /. /dev, /usr y todos los demás directorios en el directorio raíz son ramas, las cuales pueden tener sus propias ramas, como /usr/local y así sucesivamente.

Existen varias razones para albergar algunos de estos directorios en sistemas de ficheros separados. /var contiene los directorios log/, spool/ y varios tipos de ficheros temporales y pueden llegar a desbordarse. Agotar el espacio del sistema de ficheros raíz no es nada bueno desde cualquier punto de vista, así que separar /var de de / es algo que debería hacerse siempre que sea posible.

5.1. El fichero fstab

Durante el proceso de arranque los sistemas de ficheros listados en /etc/fstab se montan automáticamente (a menos que estén listados con la opción noauto).

/etc/fstab contiene una lista de líneas con el siguiente formato: *dispositivo /punto-de-montaje punto de montaje opciones dumpfreq passno*

dispositivo Un nombre de dispositivo (*debe existir*).

punto-de-montaje Un directorio (que debe existir) en el que montar el sistema de ficheros.

tipo de sistema ficheros El tipo de sistema de ficheros es un parámetro que interpretará mount. El sistema de ficheros por omisión de FreeBSD es ufs.

opciones Ya sea rw para sistemas de ficheros de lectura-escritura, o ro para sistemas de ficheros de sólo lectura, seguido de cualquier otra opción que sea necesaria. Una opción muy habitual es noauto, que se suele usar en sistemas de ficheros que no se deben montar durante la secuencia de arranque.

dumpfreq La usa para determinar qué sistema de ficheros requieren volcado. Si el campo no está declarado se asume un valor de cero.

passno

Determina el orden en el cual los sistemas de ficheros deben revisarse. Los sistemas de ficheros que hayan de saltarse deben tener su passno a cero. El sistema de ficheros raíz (que obviamente debe ser revisado antes que cualquier otro) debe tener su passno puesto a uno, y los demás sistemas de ficheros deben tener valores mayores que uno.

5.2. La orden mount

Es el comando, al fin y al cabo quien monta los sistemas de ficheros.

En su forma más básica se usa lo siguiente:

mount dispositivo punto-de-montaje

Existe una gran cantidad de opciones pero las más comunes son:

Opciones de montaje

-a

Montar todos los sistemas de ficheros que aparezcan en /etc/fstab, excepto aquellos marcados como «noauto», excluidos por el parámetro -t o aquellos que ya estén montados.

-d

Realizar todo excepto la llamada real de montaje del sistema. Esta opción es muy útil en caso de problemas si se combina con la opción -v para determinar qué es lo que [está](#) haciendo realmente.

-f

Forzar el montaje de un sistema de ficheros inestable (por ejemplo uno que da errores tras un reinicio súbito, algo que es bastante *peligroso*), o forzar la revocación de accesos de escritura cuando se cambia el estado de un sistema de ficheros de lectura-escritura a solo lectura.

-r

Montar el sistema de ficheros como sólo lectura. Esto es idéntico a utilizar el argumento ro (rdonly para versiones anteriores a FreeBSD 5.2) en la opción -o.

-t tipo de sistema de ficheros

Montar un sistema de ficheros dado con el tipo de sistema de ficheros, o montar solamente sistemas de ficheros del tipo dado si se proporciona la opción -a. «ufs» es el sistema de ficheros por omisión.

-u

Actualizar puntos de montaje en el sistema de ficheros.

-v Mostrar mayor información.

-w Montar el sistema de ficheros como lectura-escritura.

La opción -o toma una lista las siguientes opciones separada por comas:

nodev

No interpretar dispositivos especiales en el sistema ficheros. Es una opción de seguridad que puede ser muy útil.

noexec

No permitir la ejecución de binarios en este sistema de ficheros. Esta también es una opción de seguridad útil.

nosuid

No interpretar bits setuid o setgid en el sistema de ficheros. Esta también es una opción de seguridad útil.

5.3. La orden umount

Toma como parámetro un punto de montaje, un nombre de dispositivo, o la opción a o -A.

Todas las formas toman -f para forzar el desmontaje y -v para mostrar más información. Tenga muy en cuenta que usar -f no suele ser una forma recomendable de proceder. Desmontar a la fuerza los sistemas de ficheros puede acarrear el congelar la máquina o dañar los datos en el sistema de ficheros.

-a y -A se usan para desmontar todos los sistemas de ficheros montados, con la ventaja de poder elegir el tipo de sistema de ficheros que se use tras -t. De todas maneras -A no trata de desmontar el sistema de ficheros raíz.

5.4. Crear particiones con fdisk

GESTIÓN SISTEMA DE ARCHIVOS EN LINUX

Para crear **particiones** usando la herramienta **fdisk**, primero debemos comprobar los discos del sistema disponibles y las particiones que ya haya creadas, para ello utilizamos el parámetro **-l**:

```
$ sudo fdisk -l
```

```
Disco /dev/sda: 160.0 GB, 160041885696 bytes
```

```
255 cabezas, 63 sectores/pista, 19457 cilindros
```

```
Unidades = cilindros de 16065 * 512 = 8225280 bytes
```

```
Identificador de disco: 0x000c3c51
```

Dispositivo	Inicio	Comienzo	Fin	Bloques	Id	Sistema	/dev/sda2
3233	9855	53199247+	83	Linux			
/dev/sda4		9856	19457	77128065	83	Linux	

Si pulsamos la m una vez dentro podremos visualizar la ayuda, las distintas opciones con su respectiva letra de ejecución:

```
Orden (m para obtener ayuda): m
```

```
Help:
```

```
DOS (MBR)
```

- a toggle a bootable flag
- b edit nested BSD disklabel
- c toggle the dos compatibility flag

```
Generic
```

- d delete a partition
- F list free unpartitioned space
- l list known partition types
- n add a new partition
- p print the partition table
- t change a partition type
- v verify the partition table
- i print information about a partition

```
Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table
```

Vamos a crear una nueva partición, así que pulsamos “n”:

Orden (m para obtener ayuda): n

Acción de la orden

e	Partición extendida	p
Partición primaria (1-4)		

Seleccionamos si queremos una **partición extendida o primaria**, en este caso podemos crearla como primaria, pulsamos “p” y dejamos que automáticamente se configure el número de la partición (se puede especificar, del 1 al 4). Después podemos seleccionar el primer y último cilindro a utilizar para la partición. Como yo voy a usar el resto de espacio disponible será del 1 al 19457 tal y como hemos visto antes, también **podríamos indicar el tamaño de la partición en K, M o G**. Lo dejamos por defecto en este caso:

```
p
Número de partición (1-4): 3
Primer cilindro (1-19457, valor predeterminado 1):
Último cilindro, +cilindros o +tamaño{K,M,G} (1-3232, valor predeterminado 3232):
Se está utilizando el valor predeterminado 3232
```

Ahora escribimos los cambios y salimos de fdisk:

```
Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.

El núcleo todavía usa la tabla antigua.
La nueva tabla se usará en el próximo reinicio.
Se están sincronizando los discos.
```

5.5. Comando Linux para formatear particiones

Vamos a ver el comando **mkfs para formatear particiones en Linux**. Anteriormente vimos cómo crear particiones en Linux, ahora vamos a formatearlas.

En primer lugar, vamos a ver qué tipos de formatos podemos dar con mkfs. Para ello, escribimos en una consola mkfs y pulsamos tabulador, para que nos muestre las opciones disponibles:

```
root@test1:/home/tali# mkfs
mkfs      mkfs.cramfs  mkfs.ext3    mkfs.fat    mkfs.msdos  mkfs.vfat
mkfs.bfs  mkfs.ext2    mkfs.ext4    mkfs.minix  mkfs.ntfs
root@test1:/home/tali# mkfs
```

Si queremos formatear una partición de tipo FAT lo haremos con:

```
mkfs.vfat partición
```

Formatear particiones de tipo Linux. Para ello, podemos ver que los tipos de formato Linux que podemos dar son ext2, ext3 y ext4. Actualmente se está usando ext4 por lo que lo haremos con este:

```
mkfs.ext4 particion
```

5.6 Monitorizar particiones en disco Linux con comando lsblk (list block)

El comando LSBLK nos permite ver en pantalla toda la información asociada a las particiones como: Nombre, Punto de montaje, etc.

La sintaxis básica para el uso de este comando es la siguiente:

```
lsblk
```

Si deseamos obtener una información mucho más completa podemos usar el siguiente parámetro:

```
lsblk -fm
```

5.7 Monitorizar particiones en disco Linux con comando blkid (block id)

Comando que nos permite mostrar los diferentes atributos de los diversos bloques en el sistema tal como el nombre, la etiqueta y el tipo de sistema de archivos.

El parámetro básico que usaremos con blkid es el siguiente.

```
blkid
```

5.8 Monitorizar particiones en disco Linux con comando hwinfo (hardware info)

Aunque la función principal de hwinfo es mostrar información detallada acerca del hardware del equipo, podemos usar unos parámetros para visualizar un determinado valor. Es una herramienta y por tanto debe ser instalado con apt install previamente.

Por ejemplo, podemos usar el siguiente comando para desplegar el listado de los elementos de hardware como discos y particiones. Allí podemos ver las diferentes particiones y el tipo de partición.

```
hwinfo --short -block
```

5.9 Modificar la etiqueta (label) de una partición según el sistema de Archivos

Contrariamente a lo que se suele pensar, una etiqueta se aplica a una partición y no a nivel de disco o dispositivo completamente.

Es **recomendable desmontar la partición** antes de realizar ningún cambio, aunque sólo sea de etiqueta. Necesitaremos también ser `root` o usuario con permisos de administración con `sudo`.

Utilizaremos `fatlabel` si el sistema de ficheros es FAT/FAT32

```
fatlabel /dev/sdd1 nueva_etiqueta
```

Utilizamos `e2label` o `tune2fs` para el sistema de ficheros ext2/3/4

```
e2label /dev/sdd1 nueva_etiqueta
```

```
tune2fs -L nueva_etiqueta /dev/sdd1
```

En caso de tratar con NTFS utilizaremos `ntfslabel`

```
ntfslabel /dev/sdd1 nueva_etiqueta
```

5.10 Ver el tamaño de archivos y carpetas con "du"

Existen varias maneras de averiguar el peso de archivos y carpetas con este comando. A continuación, vamos a ver algunos comandos sencillos para ejecutar aplicaciones que ya vienen instaladas en todos los sistemas.

Veamos este ejemplo para saber el tamaño de un .iso o una carpeta en específico, el comando **du** se utiliza de la siguiente manera.

```
$ du -bsh /fichero_o_carpeta
```

Como vemos, se puede personalizar el uso de **du** con añadiendo algunas de estas tres opciones:

```
-b [-bytes]: Mostrar en bytes.
```

```
-s [-sumarize]: Mostrar solamente el tamaño total de cada argumento.
```

```
-h [-human-readable]: Imprime los tamaños de forma leíble (e.g., 1K, 234M, 2G)
```

5.11 Ver espacio en disco con "df"

Además del comando **du**, también existe el comando **df** para ver el uso de espacio en disco. Para utilizarlo vamos a escribir el siguiente comando:

```
$ df -h
```

Con el comando nos mostrara las particiones montadas, el uso de espacio en cada una y el espacio libre restante.