

# 104 - Dispositivos, sistemas de archivos y FHS

## 1 - Crear particiones y sistemas de archivos

### Fdisk

Fdisk es un programa para manipular particiones, para manipular una partición ejecutaremos fdisk dispositivo.

Las particiones tienen un nº hexadecimal que indica el sistema de archivos: Linux 83 (0x83) y swap 82 (0x82).

Operador	Acción
<b>-l</b>	muestra las particiones existentes y los dispositivos
<b>-n</b>	crea una partición nueva de manera interactiva
<b>-t</b>	cambia el código de codificación de la partición
<b>-d</b>	borra una partición
<b>-q</b>	sale de fdisk sin guardar la partición
<b>-w</b>	sale de fdisk y guarda las modificaciones
<b>-m</b>	muestra la ayuda

```
# Conversión de ext2 a ext3
tune2fs -j /dev/dispositivo
```

La utilidad **gdisk** es el equivalente a fdisk cuando se trata de discos particionados GPT. De hecho, la interfaz está modelada a partir de fdisk, con un indicador interactivo y los mismos (o muy similares) comandos.

### PARTED

GNU Parted es un editor de particiones muy poderoso (de ahí el nombre) que se puede usar para crear, eliminar, mover, redimensionar, rescatar y copiar particiones. Puede funcionar con discos GPT y MBR y cubrir casi todas sus necesidades de administración de discos.

Hay muchas interfaces gráficas que facilitan mucho el trabajo con parted, como GParted para entornos de escritorio basados en GNOME y el KDE Partition Manager para escritorios KDE. Sin embargo, debe aprender a usar parted en la línea de comandos, ya que en una configuración de servidor nunca puede contar con un entorno de escritorio gráfico disponible.

### Creación del sistema de archivos

Con el comando **mkfs** podemos formatear particiones, con la opción -t elegimos el sistema de archivos (ext2, ext3, ext4, xfs y vfat), por defecto sin la opción -t creará ext2.

Existen comandos específicos para cada sistema de archivos, mke2fs, nkreiserfs, mkdosfs, etc..

## Partición Swap

Para formatear la partición usaremos `mkswap /dev/partición`, una vez formateada la activaremos con `swapon -a`, que activa todas las particiones swap encontradas en `/etc/fstab`.

Para desactivarlas tenemos el comando `swapoff`

```
# Vamos a añadir swap extra
dd if=/dev/zero of=/swap bs=1M count=2048
mkswap /swap
chmod 600 /swap
swapon /swap
# es tindría que afegir la següent linea a fstab si es vol persistent:
# /swap swap swap default 0 0

# La eliminem (s'ha de treure de fstab si hi es)
swapoff /swap
rm -rf /swap
```

## 2 - Mantenimiento de la integridad de sistemas de archivos

### Chequeo sistema archivos

Para ello usamos `fsck`, para usarlo las particiones deben estar desmontadas o montadas en sólo lectura, al igual que con `fdisk` con la opción `-t` especificamos el sistema de archivos.

```
fsck -t ext4 /dev/hda
```

Opciones	Descripción
<b>-f</b>	Comprueba integridad
<b>-t</b>	tipo de partición
<b>-p</b>	intenta reparar
<b>-y</b>	respuesta a todo yes

Comando `badblock` si detectara bloques defectuosos

### Corrección del sistema de archivos

Usaremos `debugfs` para depurar los sistemas de archivos ext2 y ext3, se usa en casos que `fsck` no puede solventar el problema.

Otros comandos importantes:

- `dumpe2fs`: muestra información de grupo de bloques y superbloques

- tune2fs: configura parámetros, chequeos y propiedades de los puntos de montaje y mantenimientos automáticos (fsck)

Para xfs (sistema de archivos red hat) usamos:

- xfs\_metadump: extrae todos los datos referentes al sistema de archivos en si.
- xfs\_info: muestra características y otra información estadística sobre xfs.

### Análisis del espacio de disco

- **df**: muestra el espacio ocupado y disponible en cada dispositivo en KB, con la opción -h usa medidas como MB y GB y con la opción -T muestra el sistema de archivo.
- **du**: muestra el espacio ocupado por archivos y/o directorios, con -s puede indicarse un directorio específico. Con la opción -h usa medidas mas legibles.

## 3 - Control y montaje de los sistemas de archivos

### Fstab

Todos los sistemas de archivos creados en la instalación de Linux se montarán automáticamente en el inicio del sistema.

La información se almacenará en /etc/fstab.

Es necesario que exista una linea para cada sistema de archivos en fstab, en este archivo se determinan las particiones, puntos de montaje, etc.

Los elementos de cada linea son:

- Partición del dispositivo
- Punto de montaje
- Tipo de sistema de archivos
- Opciones de montaje
  - rw: lectura y escritura
  - ro: sólo lectura
  - noauto: no montar automáticamente
  - users: los usuarios la podrán montar y desmontar
  - owner: los permisos del dispositivo se adecuarán al usuario que lo montó
  - usrquota: activa el uso de cuotas de usuario
  - grpquota: activa el uso de cuotas de grupos
  - remount: vuelve a montar el dispositivo
- Dump (es un 0 o 1 al final de la linea): 0 indicará apagado, 1 funcionando.
- Fck (es un 0 o 1 al final de la linea): 0 indicará apagado, 1 funcionando.

Ejemplo de fichero fstab usado con identificación de partición UUID (podemos conocer el UUID con el comando blkid) y diferentes tipos sistemas de archivos:

```
# SWAP /dev/sdb2
UUID=c0e80b01-911b-4670-ae80-6ba22a35c027      none      swap      defaults
0 0

# /dev/sdb3
UUID=aedcefe1-d2e2-4a3e-ab11-d8b596a60f9b      /          ext4      rw,relatime
0 1

# /dev/sdb1
UUID=8A72-37FB /boot/efi      vfat
rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,
utf8,errors=remount-ro 0 2

# /dev/sdb4
UUID=dfab7cc3-4d23-4d46-9fcc-775421103e21      /home      ext4      rw,relatime
0 2

# /dev/sdc2
UUID=B8E08296E0825B10 /run/media/Documentos ntfs
defaults,user,uid=1000,gid=1000,umask=022 0 0

# /dev/sdd1
UUID=69a04974-9ea2-43b4-acdb-a7ed22534bd8 /run/media/Games ext4 defaults
0 0
```

## Montaje manual de sistemas de archivos

El comando **mount** si no se usa con argumentos, mostrará los dispositivos montados y el detalle de los mismos.

```
# Así montaremos todo lo que haya en fstab que no esté marcado como noauto
mount -a

# Uso habitual de mount:
mount [opciones] [--source] <fuente> | [--target] <directorio>

# Desmontaje de todos los sistemas de ficheros
umount -a

# Desmontaje con umount
umount [opciones] <origen>
```

## 4 - Administrar cuotas de disco

Es posible limitar la cantidad de espacio de disco a un determinado usuario o grupo, para activar el control de cuotas, es necesario que el kernel soporte esta función y se incluya la opción **usrquota** o **grpquota** en **/etc/fstab**

Para generar la tabla de estadísticas de uso de disco, usamos `quotacheck -a`.

Para crear configuraciones de cuota usamos `edquota`, con `-u` modificamos para usuario y con `-g` para grupo.

La configuración de cada partición se almacena en raíz del mismo y son `aquota.user` o `aquota.group`.

Para que las cuotas pasen a monitorearse y controlarse usamos `quota -a`.

Es posible avisar al usuario cuando ha alcanzado su límite con `edquota -ta`.

El usuario root puede generar informes de cuota con `repquota -a`.

## 5 - Controlar permisos y propiedades de los archivos

Tenemos un sistema permisos de archivos y directorios, existen tres niveles de permisos:

- Dueño del archivo (u)
- Grupo del dueño del archivo (g)
- Otros (o)

Los permisos podemos ver con `ls -l`.

La primera letra indica el tipo de archivo que estamos tratando:

- `d` : directorio
- `l` : enlace simbólico
- `c` : dispositivo especial de caracteres
- `p` : canal fifo
- `s` : socket
- `-` : archivo convencional

Las demás letras se agrupan en grupos de tres determinando el resto de permisos.

### Modificación de permisos

Para modificar los permisos usamos `chmod`:

```
# Le quitamos permisos a otros de lectura, al usuario permisos r-x y al grupo
le damos permisos de lectura.
>> .rw-rw-rw- ziko ziko 0 B Tue Mar 12 18:17:33 2024 texto.txt

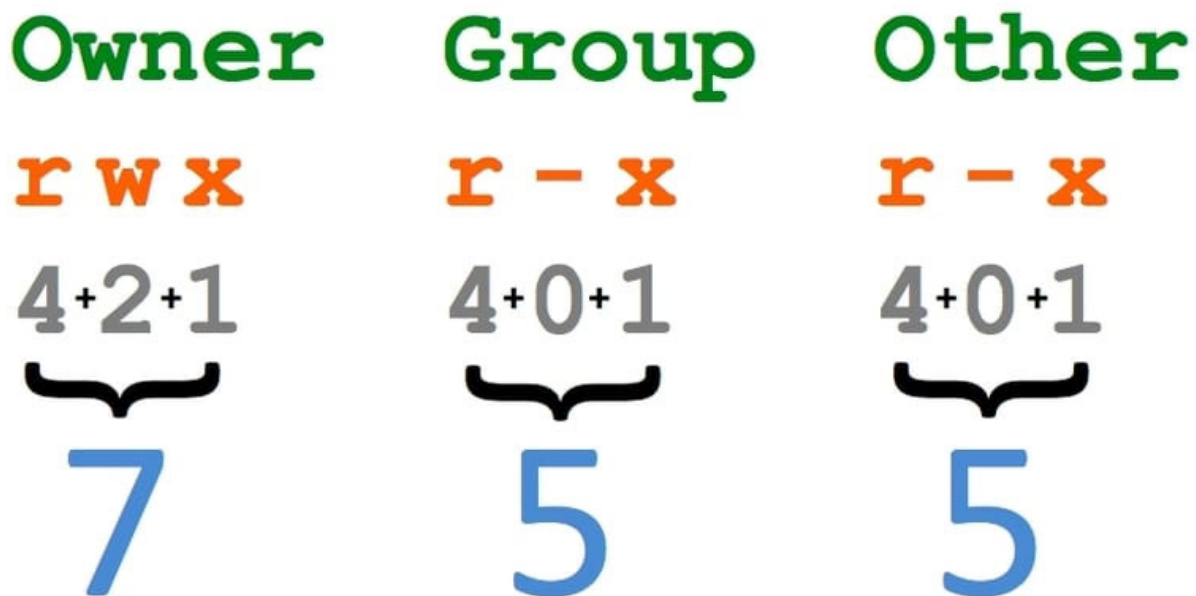
chmod u=r-x,g=r,o-r texto.txt

>> ls -l texto.txt
>> .r-xr---w- ziko ziko 0 B Tue Mar 12 18:17:33 2024 texto.txt
```

```
# Damos permisos de escritura al grupo
chmod g+w texto.txt
```

En los directorios, los permisos son un poco diferentes, estos son los significados reales:

- r : permite acceder
- w : permite crear/copiar archivos dentro
- x : permite listar el contenido en él



This image is part of the Bioinformatics Web Development tutorial at [http://www.cellbiol.com/bioinformatics\\_web\\_development/](http://www.cellbiol.com/bioinformatics_web_development/)  
© cellbiol.com, all rights reserved

## Permisos octales

Es mas eficiente manejar con permiso ocales, van en orden usuario, grupo y otros.

Dígito	Lectura (valor 4)	Escritura (valor 2)	Ejecución (valor 1)
0 (000)	-	-	-
1 (001)	-	-	x
2 (010)	-	w	-
3 (011)	-	w	x
4 (100)	r	-	-
5 (101)	r	-	x
6 (110)	r	w	-
7 (111)	r	w	x

```
# Permisos u (4+2 RW), g (4+2 RW), o (4 R)
chmod 664 texto.txt
```

```
>> ls -l texto.txt
>> .rw-rw-r-- ziko ziko 0 B Tue Mar 12 18:28:58 2024 texto.txt
```

## Umask

Es un filtro para la creación de nuevas carpetas y archivos, lo mas habitual es el 022.

Para conocer el que tenemos en nuestro sistema deberemos usar el comando `umask` sin ningún argumento.

Las carpetas tienen por permiso 777 y los archivos 666 en el caso de nueva creación, al aplicar el filtro umask, las carpetas quedarían como permisos 755 y los archivos 644.

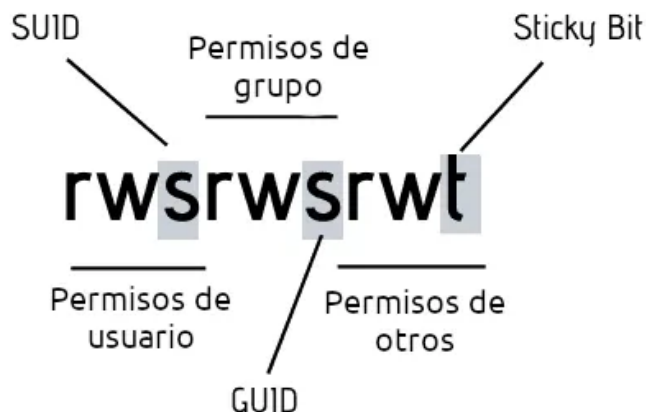
Para modificarlo, simplemente tipeamos `umask nuevo_num_umask`, pero para que sea permanentemente debe ir en `/etc/profile` o en el `.bashrc` de cada usuario según nos interese.

## Setuid, sguid y sticky bit

- *Setuid*:
  - Se trata de un fichero al que daremos permisos de ejecución como si fuera root, pero podrá ser ejecutado como un usuario, el clásico ejemplo es el comando `passwd`, que permite el cambio de su password a un usuario.
  - La búsqueda con find será con `-perm -4000`
  - `chmod u+s fichero -- chmod 4xxx fichero`
  - Se identifica viendo una s en el puesto de la x en usuario
    - `·rwsr--r--`
- *Setgid*
  - Se trata de lo mismo que setuid pero en este caso en una carpeta, una carpeta con permisos root, pero que puede ser modificada por usuarios normales (útil cuando muchos usuarios trabajan en un mismo grupo)
  - La búsqueda con find será con `-perm -2000`
  - `chmod g+s my_dir/ -- chmod 2xxx my_dir/`
  - Se identifica viendo una s en el puesto de la x de grupo
    - `drwxrwsr--`
- *Sticky bit*
  - Una carpeta a la que pueden acceder todos los usuarios, pero sólo root y el usuario creador del archivo puede borrar dentro de esa carpeta, no se podrán borrar ficheros de otros usuarios (usado en la carpeta `/tmp`).
  - La búsqueda con find será con `-perm -1000`
  - `chmod +t my_dir/ -- chmod 1xxx my_dir/`
  - Se identifica con una t en el último de los permisos
    - `drwxrwxrwt`

[Explicación ampliada](#)

Imagen permisos:



## Modificar dueños y grupos de archivos

```
# Para cambiar el usuario
>> .rw-r--r-- ziko ziko 0 B Tue Mar 12 18:32:00 2024 test
chown mariano texto.txt
>> .rw-r--r-- mariano ziko 0 B Tue Mar 12 18:32:00 2024 test

# Para cambiar el grupo siguiendo el ejemplo anterior
chgrp mangotero texto.txt
>> .rw-r--r-- mariano mangotero 0 B Tue Mar 12 18:32:00 2024 test

# Para cambiar usuario y grupo a la vez
chown taco.bell test
>> .rw-r--r-- taco bell 0 B Tue Mar 12 18:32:00 2024 test

# Si añadimos la opción -R se modificaran todos los demás archivos
repetidamente
```

## 6 - Archivos de sistema, ordenación y localización

FHS (Filesystem Hierarchy Standard): es el estándar de localización adoptado por casi todas las distros, cada directorio tiene un propósito.

### Directorios que residen obligatoriamente en raíz

Directorio	Contenido
<b>/bin y /sbin</b>	Contienen los programas para cargar el so y comandos especiales
<b>/etc</b>	Archivos de configuración específicos del so
<b>/lib</b>	Bibliotecas compartidas por /bin, /sbin y módulos de kernel
<b>/mnt y /media</b>	Puntos de montaje
<b>/proc y /sys</b>	Directorios con información de procesos y hardware
<b>/dev</b>	Archivos de acceso a dispositivos y archivos especiales



## Directorios que pueden ser puntos de montaje

Directorio	Contenido
<b>/boot</b>	Kernel, mapas del sistema y cargadores de boot
<b>/home</b>	Directorios de los usuarios
<b>/root</b>	Directorio del usuario root
<b>/tmp</b>	Archivos temporales
<b>/usr/local y /opt</b>	Programas adicionales y bibliotecas
<b>/var</b>	Datos de programas

## Localización de archivos

Además de find podemos usar **locate**, es más rápida que find, pues se busca en una base de datos y no en el disco.

Esta base de datos debe actualizarse regularmente con el comando **updatedb**

El archivo `/etc/updatedb.conf` contiene que directorios y sistemas de archivos debe ignorar la actualización de la base de datos.

El comando **which** se usa para que nos devuelva la ruta absoluta al programa solicitado buscando en los directorios de la variable PATH.

Con el comando **whereis**, nos devuelve las rutas a los archivos ejecutables, código fuente y página de manual.