

---

# Gestión de discos. Sistemas de ficheros

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.escet.urjc.es>



Curso 2005–2006

---



## Organización de ficheros

- Árbol jerárquico. Carácter “/” para recorrer el árbol de la raíz hasta las hojas (“padre/hijo”).
- La estructura del árbol está compuesta de *directorios*, que son los únicos que pueden tener ficheros debajo, y el resto, que son las hojas del árbol.
- Todos los directorios contienen al menos dos ficheros:
  - “.”: representa al propio directorio.
  - “..”: enlace a su directorio padre.
- Raíz: directorio “/”. El padre del directorio raíz es el mismo directorio raíz.
- Cada directorio, y todo su contenido, puede pertenecer a un “punto de montaje” ( $\simeq$  disco) distinto.

## Acceso a los ficheros

- Todo proceso tiene un directorio de trabajo preferente, el “directorio actual” (orden “`pwd`” de la *shell*).
- Acceso a ficheros, mediante un *camino* (lista de nodos separados por “/”).
  - Si el *camino* comienza por “/”, indica que se debe recorrer el árbol desde la raíz.
  - Si comienza por cualquier otro carácter, se recorre desde el directorio actual.

Ejemplos: “ `practica/f1.txt` ”

“ `/home/./home/cespedes/practica/f1.txt` ”

“ `./././././././tmp/./etc/passwd` ”

## Puntos de montaje

- Normalmente, no todos los ficheros del árbol de directorios se encuentran en el mismo disco.
- *Punto de montaje*: directorio que pertenece a un disco (o *partición*) distinto, junto con todo su contenido (excluyendo otros puntos de montaje).
- El *kernel* incluye información de todas las particiones y puntos de montaje asociados en el fichero `/proc/mounts` (el fichero `/etc/mtab` contiene prácticamente la misma información).
- También se pueden consultar los puntos de montaje junto con los discos o particiones que están *montadas* en ellos con las órdenes “`mount`” y “`df`”.

## mount, df

- **mount**: Muestra las particiones, puntos de montaje, tipo de partición y opciones de cada una de ellas:

```
/dev/hda2 on / type ext3 (rw,noatime)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda5 on /scratch type ext3 (ro,noatime)
tmpfs on /tmp type tmpfs (rw)
```

- **df**: Muestra cada una de las particiones *con ficheros reales* montadas en el sistema, el punto en el que está montada, su capacidad y su uso:

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/hda2	28842780	6957692	20419960	26%	/
/dev/hda5	38448276	32838556	3656620	90%	/scratch
tmpfs	517960	1196	516764	1%	/tmp

## Montaje de sistemas de ficheros ya existentes

- Crear el directorio si no existe:  
`mkdir /var`
- Hacer visible el sistema de ficheros bajo ese directorio:  
`mount -t ext2 -o rw /dev/hda3 /var`
- Si queremos desmontar (o hacer invisible) un sistema de ficheros que esté montado en el directorio `/var`:  
`umount /var`

## /etc/fstab

- Automatización del montaje de sistemas de ficheros en el arranque

```
# <filesystem> <mount point> <type> <options> <dump> <pass>
proc           /proc         proc         defaults    0          0
/dev/hda2     /             ext3        noatime     0          1
/dev/hda5     /scratch     ext3        noatime,ro  0          1
/dev/hda6     none         swap        sw          0          0
tmpfs         /tmp         tmpfs       defaults    0          0
```

- La columna “dump” indica si esta partición se va a incluir en el sistema automático de *back-up*, y la columna “pass” indica el orden en el que se va a comprobar la integridad de las particiones en el arranque (0: desactivado).
- Montaje automático del contenido de `/etc/fstab` con la orden `mount -a`.

## Tipos de sistemas de ficheros

- Los que reconoce el *kernel* están en `/proc/filesystems`.
- “Tradicionales”:
  - **msdos**: El usado por MS-DOS y Windows pre-95, sin permisos ni dueños, nombres de fichero de 8 caracteres con extensiones de 3 caracteres.
  - **vfat**: Usado a partir de Windows-95, compatible con MS-DOS pero con posibilidad de nombres de fichero largos.
  - **ntfs**: A partir de Windows NT, añade características de seguridad (permisos, dueños, etc).
  - **iso9660**: Sistema de fichero utilizado en los CDs de datos.
  - **minix**: usado por MINIX y por los primeros Linux.
  - **ext2**: Sistema de ficheros por excelencia en Linux.

## Tipos de sistemas de ficheros (2)

- Con “*journal*”:
  - **ext3**: Siguiete versión del **ext2**, idéntico pero con adición de *journal*. El más utilizado hoy en día.
  - **reiserfs**: Primer sistema de ficheros con *journal* para Linux; muy eficiente con directorios grandes.
  - **jfs**: Sistema de ficheros con *journal* de IBM.
  - **xfs**: Sistema de ficheros con *journal* creado por SGI para su plataforma IRIX.

## Tipos de sistemas de ficheros (3)

- Con características especiales:
  - **romfs**: ideado para discos pequeños y de solo lectura, típicamente discos grabados en ROM o en EPROM.
  - **cramfs**: similar al **romfs**, pero incluyendo compresión de los datos.
  - **autofs**: pseudo-sistema de ficheros para permitir el auto-montaje de unidades extraíbles.
  - **umsdos**: Compatible con discos MS-DOS, añade características esenciales para sistemas UNIX (dueños y grupos, permisos, nombres largos, enlaces, etc).

## Tipos de sistemas de ficheros (4)

- No asociados a dispositivo:
  - **proc, sysfs**: Información proporcionada por el *kernel* a los procesos de usuario.
  - **devfs, devpts**: Sistema de ficheros en el que se generan automáticamente los dispositivos reconocidos por el *kernel* en cada momento.
  - **tmpfs, ramfs**: Sistema de ficheros de lectura/escritura, con contenido en memoria, con carácter temporal y para permitir un acceso más rápido.
  - **usbfs**: lista de dispositivos conectados al *bus* USB y características de cada uno de ellos.

## Tipos de sistemas de ficheros (5)

- Remotos:
  - **nfs**: *Network File System*, desarrollado por SUN, el más usado entre los sistemas ficheros remotos en UNIX. proporcionada por el *kernel* a los procesos de usuario.
  - **smb, cifs**: Sistemas de ficheros remotos usados por entornos Windows.
  - **ncp**: *Netware Core Protocol*, protocolo sobre IPX para montar sistemas de ficheros de Novell Netware.
  - **coda**: Sistema de ficheros remoto avanzado, con muchas ventajas sobre NFS, pero muy complicado de poner en funcionamiento.
  - **afs**: *Andrew filesystem*.

## Tipos de sistemas de ficheros (6)

- Además de los mencionados, en Linux hay soporte para muchos otros sistemas de ficheros, la mayor parte de ellos utilizados por otras plataformas para guardar su información.
- Entre ellos están: **udf**, **adfs** (*Acorn Disc Filing System*), **affs** (*Amiga Fast File System*), **asfs** (*Amiga Smart File System*), **hfs** (Apple Macintosh), **befs** (BeOS), **bfs** (*Boot File System*, SCO), **efs** (SGI, IRIX), **jffs** (*Journaling Flash File System*), **vxf**s (VERITAS VxFS), **hpfs** (OS/2), **qnx4**, **sysv** (System V), **ufs** (SunOS, FreeBSD, NetBSD, OpenBSD)...

## Opciones de los sistemas de ficheros

- **defaults**: Ninguna opción especial.
- **rw**: Permisos de lectura y escritura.
- **ro**: Sólo lectura.
- **loop**: Permite montar un *fichero* en lugar de un dispositivo de bloques.
- **noauto**: No montar automáticamente con “**mount -a**” (esto es, en el inicio).
- **user**: Puede montarlo y desmontarlo un usuario distinto de **root**.
- Para otras opciones, consultar la página de manual de **mount**.

## Cómo añadir un disco al sistema

Para añadir un disco nuevo al sistema, es necesario:

1. Preparar las particiones del disco
2. Crear el sistema de ficheros
3. Elegir (y crear) el punto de montaje
4. Montar el nuevo disco,
5. Añadir contenido al disco.
6. Modificar `/etc/fstab` para que se monte automáticamente en el arranque.

## 1. Preparar las particiones del disco

- No siempre es necesario: en disquetes no se utilizan nunca, y en llaveros USB, a veces tampoco se hacen particiones y se utiliza todo el disco (`/dev/sda` en lugar de `/dev/sda1`).
- Uso de las órdenes “`fdisk`”, “`cdisk`” o “`sdisk`” para modificar la tabla de particiones.
  - `fdisk`: Herramienta original, es necesario hacer todo “a mano”, algo complicada.
  - `cdisk`: Con menús a toda pantalla, bonito, fácil de usar.
  - `sdisk`: Orientado para ser usado desde *scripts*.

## 2. Crear el sistema de ficheros

- Hay que elegir el tipo de sistema de ficheros más adecuado.
- Normalmente, a menos que haya necesidades especiales, se utilizará el mismo sistema para todas las particiones. Se suele elegir “**ext3**”, o bien algún otro sistema con *journaling*.
- Creación del sistema de ficheros con la orden “**mkfs**”:  

```
mkfs -t ext3 /dev/hda3
```

## 3. Elegir (y crear) el punto de montaje

- Los puntos de montaje más habituales, entre los directorios “estándar”, son: **/usr**, **/var**, **/home**, **/tmp**, **/var/tmp**.
- Pero se puede crear cualquier otro, dependiendo de nuestras necesidades, por ejemplo **/musica**.
- Antes de poder *montar* un nuevo sistema de ficheros en un directorio, es necesario que este directorio exista: hay que crearlo con **mkdir**.
- Los permisos que tenga este directorio no son muy relevantes, ya que desaparecerán una vez montado el nuevo sistema de ficheros.

## 4. Montar el nuevo disco

- Si queremos montar un disco de manera temporal, para examinarlo o para comprobar que ha sido creado correctamente, se suele hacer en un directorio especial para montajes temporales, el `/mnt`:

```
mount -t ext3 /dev/hda3 /mnt
```

## 5. Añadir contenido al disco

- El contenido original del punto de montaje, al igual que los permisos que tenga, desaparecerán una vez montado el nuevo sistema de ficheros.
- Si queremos que el nuevo disco tenga la misma información que tenía ese directorio, hay que hacer una copia (con `cp`, `rsync...`):

```
rsync -avHWSx /var /mnt
```

## 6. Modificar `/etc/fstab`

- Por cada punto de montaje que tengamos en el sistema, es necesaria una línea nueva en el fichero `/etc/fstab`.
- Normalmente, las opciones no se modifican (“`defaults`”).
- El contenido original del punto de montaje, al igual que los permisos que tenga, desaparecerán una vez montado el nuevo sistema de ficheros.
- Si queremos que el nuevo disco tenga la misma información que tenía ese directorio, hay que hacer una copia (con `cp`, `rsync...`)

## Ejemplo de montaje de un nuevo disco

Estas acciones deberán realizarse preferiblemente en modo mono-usuario, para evitar que haya procesos que interfieran con nuestras acciones.

Si queremos crear una nueva partición, `/dev/hda3`, para montarla en el directorio `/var`:

```
# cfdisk /dev/hda
# mkfs -t ext3 /dev/hda3
# mount -t ext3 /dev/hda3 /mnt
# rsync -avHWSx /var /mnt
# rm -rf /var
# mkdir /var
# umount /mnt
# echo "/dev/hda3 /mnt ext3 defaults 0 2" >> /etc/fstab
# mount /var
```

## Integridad del sistema de ficheros

- En el arranque, se hace una comprobación rutinaria de los sistemas de ficheros, buscando fallos que puedan ser corregidos.
- Esta comprobación se realiza con la orden “**fsck**”, y puede realizarse a mano (siempre con el disco sin montar):

```
fsck -t ext3 /dev/hda3
```