

Administración de discos

Joaquín Seoane Pascual

joaquin@dit.upm.es

Departamento de Ingeniería de Sistemas Telemáticos

Universidad Politécnica de Madrid

5 de noviembre de 2001

Índice General

Los discos	3
Sistemas de ficheros	7
Discos de Win/DOS	16
Sistemas de seudoficheros	17
Paginación	18
Fichero de definición de sistemas de ficheros	19
Organización de los sistemas de ficheros	22

Los discos

- Guardan datos *constantes* y *variables*.
- Dimensionar en capacidad, velocidad, número y tolerancia a fallos.
- Puede ser conveniente partirlos (en *particiones*) para:
 - Adaptar al uso.
 - Permitir protección selectiva.
 - Permitir reparación selectiva.
 - Evitar invasiones.
 - Facilitar copias de seguridad.
 - Mejorar rendimiento.

Los discos y las particiones en Linux

Aparecen como dispositivos en el sistema de ficheros:

- `/dev/hda`: disco 1 IDE/ATAPI.
- `/dev/hdb`: disco 2 IDE/ATAPI (o cederrón).
- `/dev/sda`: disco 1 SCSI.
- `/dev/sdb`: disco 2 SCSI.
- `/dev/hda1`: partición 1 de disco 1 IDE/ATAPI.
- `/dev/sda1`: partición 1 de disco 1 SCSI.
- `/dev/fd0`: disquete ordinario.
- `/dev/ram0`: Disco RAM.

Las particiones

- Tabla al comienzo del disco.
- Varios convenios: MS/DOS, BSD, SunOS, Solaris, Ultrix,...
- MS/DOS original: 4 particiones (*primarias*)
- MS/DOS ampliado: 1 partición puede ser extendida y tener varias lógicas.

Volúmenes lógicos

Se montan encima de volúmenes físicos o ficheros.

- `/dev/loop0`: Simulación de volumen en fichero:
 - `losetup /dev/loop0 fichero`
- `/dev/md0`: RAID (metadisco) construido con discos físicos.
 - Lineal: se usa uno tras otro, para aumentar capacidad..
 - Raid 0: se usan en paralelo, para aumentar velocidad y capacidad.
 - Raid 1: espejo, para aumentar fiabilidad (y velocidad).
 - Raid 5: tolerante a fallo de un disco (capacidad = $(N - 1) \times C$).
- `/dev/vg/v1`: LVM (gestión de volúmenes lógicos), para gestionar dinámicamente.

Sistemas de ficheros

- Son formas de organizar los datos en un disco (o protocolos de acceso remoto).
- Tienen propiedades diversas:
 - Longitud de nombres.
 - Sistema de protección, propietarios,...
 - Tipos (ficheros, directorios, dispositivos, pipes, sockets, enlaces simbólicos, ...).
 - Atributos (tiempos, seguridad, compresibilidad, archivabilidad, inmutabilidad, ocultación, ...)

Sistemas de ficheros en Linux

Un sistema abierto debería soportar varios:

- `ext2`
- `minix`
- `ufs`
- `msdos`, `vfat`, `umsdos`
- `hpfs`
- Transaccionales (`xf`s, `rieserfs`, `jfs`, `ext3`), ...
- `iso9660` (Rock-Ridge, Joliet) (CD-ROM)
- ...
- Remotos: `nfs`, `smb`, `afs`, `coda`, `ncp`, ...

Creación de sistemas de ficheros

- `format c:` en MS-DOS
- `mkfs` en Unix (general)
- `mke2fs` o `mkfs.ext2` en Linux (ext2)
- `mkfs.minix` en Linux (minix)
- Ejemplos:
 - En dispositivo: `mkfs.ext2 /dev/hda1`
 - En fichero:

```
dd if=/dev/zero of=fichero bs=1024 count=65536  
mkfs.ext2 fichero
```

Montaje de sistemas de ficheros

- `mount -t ext2 /dev/hda2 /usr`
- `mount -t vfat /dev/fd0 /mnt`
- El tipo de sistema de ficheros a veces puede descubrirse *mágicamente*:
`mount /dev/fd0 /mnt`
- Los montajes pueden tener parámetros específicos:
`mount -o ro,noexec,nosuid,sync /dev/fd0 /mnt`
- De fichero:
`losetup /dev/loop0 fichero`
`mount /dev/loop0 /mnt`
`0`
`mount -o loop fichero /mnt`

Incompatibilidades por defecto

- No todos aceptan las mismas operaciones. Ejemplo:

```
mount -t msdos /dev/fd0 /mnt
cd /mnt
touch nombre_muy_largo_para_dos
ls
chown joaquin nombre_m
ln -s nombre_m cosa
umount /mnt
```

- Se hace corresponder a la semántica de Unix lo máximo posible.

Incompatibilidades por exceso

- Órdenes y llamadas especiales para atributos especiales:
 - lsattr y chattr para ext2:
 - A** No modificar tiempo de acceso.
 - a** Sólo se puede añadir.
 - c** Comprimido.
 - i** Inmutable.
 - s** Se pone a ceros al borrar.
 - S** Síncrona.
 - d** No se hace copia de seguridad (dump).
- Copias de seguridad especiales.

Parámetros de los sistemas de ficheros

- Normalmente se pasan al crear:
 - Número de i-nodos, tamaño de bloque
(ej: `mkfs.ext2 -b 1024 -N 20000 /dev/hda1`):
 - Tamaño máximo de nombres
(minix, ej: `mkfs.minix -l 14 /dev/fd0`).
- A veces se pueden modificar (`tune2fs`):
 - Porcentaje reservado para root.
 - Número de montajes sin comprobación.

Comprobación de coherencia y reparación

- scandisk en Windows.
- fsck en Unix (general).
- e2fsck o fsck.ext2 en Linux (ext2).
- fsck.minix en Linux (minix).
- fsck.reiserfs en Linux (reiserfs).
- fsck.msdos en Linux (msdos).

Control de ocupación

- Todos los sistemas de ficheros montados: `df`.
- Uno: `df /dev/hda2`.
- Inodos: `df -i /dev/hda2` (sin sentido en `vfat`).

Discos de Win/DOS

- Pueden montarse.
- Por comodidad se pueden manejar desmontados con el paquete `mtools`.
 - `mdir a:`
 - `mcopy a:* .`
 - `mcopy * a:`
 - `mmd a:cosa`
 - `mcd a:cosa`

Sistemas de seudoficheros

- Interesa uniformizar la forma de acceso a cualquier información
- El sistema de ficheros `proc` en `/proc`.
 - `cpuinfo`
 - `ioports`
 - `interrupts`
 - `meminfo`
 - Un directorio por proceso activo
- El sistema de ficheros `devpts` para seudotermiales.
- El sistema de ficheros `devfs` para sustituir a `/dev`.

Paginación

- Para memoria virtual que no cabe en la principal.
- Partición o fichero.
- Conviene en partición (una por disco).
- Operación de creación en partición (da formato): `mkswap /dev/hda2`.
- Operación de creación en fichero:
`dd if=/dev/zero of=swapfile bs=1024 count=65536`
`mkswap swapfile`.
- Operación de activación: `swapon /dev/hda2`.

Fichero de definición de sistemas de ficheros

- /etc/fstab.
 - Qué se monta al arrancar y dónde.
 - Parámetros de montaje (en arranque y posteriores).
 - Qué sistemas de ficheros se comprueban y en qué pasada.
 - Qué sistemas de ficheros se respaldan con dump.
 - Dónde paginar.

Ejemplo de /etc/fstab

```
/dev/hdb1 /          ext2      defaults,errors=remount-ro 0 1
/dev/hdb2 none      swap      sw        0 0
proc      /proc     proc      defaults  0 0
/dev/hdb3 /usr      ext2      defaults  0 2
/dev/hda3 /home     ext2      defaults  0 2
/dev/hda1 /dos      vfat      uid=1013  0 0
/dev/hdd  /cdrom   iso9660   user,ro,noauto 0 0
/dev/fd0  /fd      vfat      uid=1013,user,noauto 0 0
```

Montajes basados en /etc/fstab

- `mount -a`
- `mount -a -t vfat`
- `mount /usr`
- `mount /dev/hda3`

Organización de los sistemas de ficheros

- Facilita la gestión de discos:
 - Ubicando ficheros por tipos.
 - Facilitando su reparto en volúmenes.
 - Facilitando el proceso de arranque.
 - Facilitando las copias de seguridad.
 - Facilitando la restauración de un sistema configurado.

El FHS

- Intento de unificación de sistemas Unix libres.
- El árbol de directorios se construye montando sistemas de ficheros.
- Separar cosas distintas en directorios y sistemas de ficheros.
- La raíz debe tener lo necesario para arrancar, montar y reparar y ser pequeña y relativamente poco activa.

La raíz

- `/bin` para binarios básicos.
- `/sbin` para binarios de administración.
- `/lib` para bibliotecas dinámicas.
- `/lib/modules` para módulos del núcleo.
- `/etc` para configuración.
- `/dev` para dispositivos.
- `/proc` para información del núcleo.
- `/tmp` para temporales.
- `/boot` para arranque.
- `/root` para cuenta del administrador.

Compartibles modo lectura

- Misma arquitectura:
 - `/usr/{bin,sbin,lib}` para la distribución.
 - `/usr/local/{bin,sbin,lib}` para cosas locales.
- Todas las arquitecturas:
 - Deberían ir en `/usr/share/{man,info,doc}`

Escribibles

- No compartibles
 - /var/log para registros.
 - /var/spool para colas.
 - /var/run para PIDs.
 - /var/lock para cerrojos.
 - /var/preserve para respaldos.
 - /var/tmp para temporales.
 - /var/cache para caches.
- Compartibles
 - /home para cuentas.