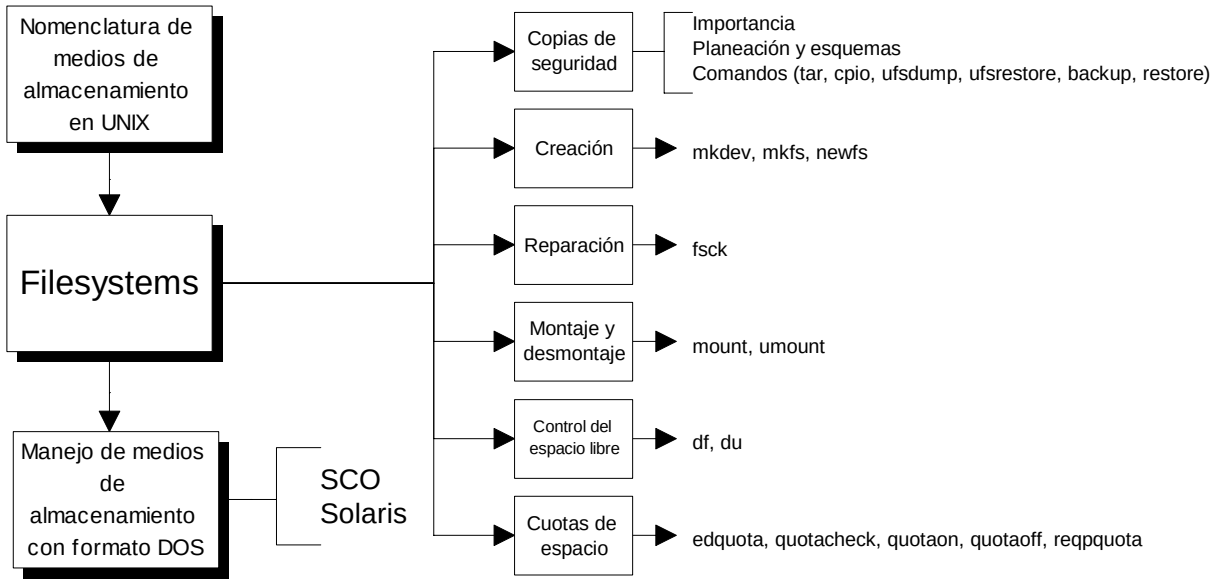
ADMINISTRACION DE DISPOSITIVOS DE ALMACENAMIENTO

En el presente capítulo se seguirá el siguiente esquema de estudio:



INTRODUCCION

Los medios de almacenamiento son de capital importancia para cualquier sistema operativo, por dos razones:

- Almacenan los programas que componen el sistema operativo, y los programas de aplicación del usuario.
- Almacenan los datos del usuario de manera no volátil, actuando como soporte de la memoria RAM.

En los inicios de la informática, los medios de almacenamiento más populares fueron las tarjetas y cintas perforadas. En la medida en que los medios magnéticos rebajaron sus costos, se volvieron más populares que los primeros y los reemplazaron. En la época actual ha habido notables avances en las tecnologías de almacenamiento óptico, al tiempo que los medios magnéticos tienen cada vez más capacidad y son más compactos.

Los soportes magnéticos son preferidos en este momento para la operación rutinaria de los sistemas de información, debido a su rapidez. Los medios ópticos se emplean preferentemente para hacer copias de seguridad y conservar datos fuera de línea, debido a su alta capacidad y a que no son tan rápidos como los magnéticos.

NOMENCLATURA DE LOS DISPOSITIVOS DE ALMACENAMIENTO EN UNIX

Dependiendo de la versión de UNIX, los dispositivos de almacenamiento tendrán nombres diferentes. No obstante, debe recalcarse que la filosofía de UNIX se aplica a

todo el sistema: Cada dispositivo es representado por un archivo, que aparece por lo general en el directorio `/dev`.

Discos duros y filesystems

Bajo SCO UNIX, existen dos archivos de dispositivo por cada división que haya en un disco duro. Dichas divisiones se crean con el comando `divvy` en el momento de la instalación del sistema operativo o de un nuevo disco duro. A cada una de ellas se le da un nombre (por ejemplo, `root` ó `u`) y por lo general en cada una de ellas se crea un filesystem.

Cada uno de estos filesystems (y todos los medios magnéticos) se pueden acceder de dos maneras diferentes:

- *Modalidad bloque*: En esta modalidad, los datos se transfieren a través de un buffer desde y hacia el dispositivo. Esto exige que las transferencias se efectúen en bloques de datos del tamaño del buffer.
- *Modalidad RAW (o ráfaga)*: En esta modalidad, los datos se transfieren byte por byte sin pasar por el sistema de buffers.

Para el ejemplo anterior, se crearían los archivos `/dev/root` y `/dev/u`, que permiten acceder a los filesystems en modalidad bloque; y otros dos llamados `/dev/rroot` y `/dev/ru`, que permiten acceder a ellos en modalidad raw. Esta convención se cumple para todas las versiones de UNIX: Cada dispositivo tiene un nombre básico que permite acceder a él en modalidad bloque. Para accederlo en modalidad raw, se construye el nombre del dispositivo anteponiéndole una `r`.

Bajo Solaris, el nombre de los dispositivos relacionados con los filesystems tiene que ver con la manera como las máquinas Sun manejan los discos:

- Los discos duros de una máquina Sun emplean la tecnología SCSI (Small Computer Systems Interface). Esta tecnología permite conectar diversos dispositivos, como discos duros, scanners, unidades de CD-ROM y muchos otros, a un mismo bus. Un computador puede tener varias tarjetas controladoras SCSI; cada una de ellas puede soportar la conexión de hasta 7 periféricos. Cada tarjeta controladora está identificada con un número único, iniciando con el 0.
- Cada uno de los periféricos conectados a una controladora tiene un número de identificación único que va de 0 a 6.
- Cada uno de los periféricos posee a su vez uno o varios números lógicos de unidad (LUN, Logical Unit Number). En la mayoría de los casos, cada periférico tiene sólo un LUN (el 0). Sin embargo, periféricos complejos como las rocolas de CD-ROM y los arreglos de discos disponen de varios LUNs para así poder identificar los periféricos individuales que hay dentro de él (un CD-ROM en particular, por ejemplo).
- En el caso particular de los discos duros, cada uno de ellos puede ser particionado hasta en 8 "tajadas", donde cada una de ellas puede contener un filesystem. Cada una de ellas posee un número de identificación, del 0 al 7.

Hechas estas consideraciones, se puede especificar el formato que tienen los nombres de filesystems en Solaris:

`/dev/dsk/c[`*`controladora`*`]t[`*`identificador SCSI`*`]d[`*`LUN`*`]s[`*`número tajada`*`]`

Por ejemplo, el filesystem root de una máquina Sun se llama por lo general:

```
/dev/dsk/c0t3d0s0
```

Para accederlo en modalidad raw, se emplea el siguiente nombre de dispositivo:

```
/dev/rdsk/c0t3d0s0
```

Disquetes

En SCO UNIX, las unidades de disquete se acceden en modo bloque de la siguiente manera:

`/dev/fd[`*`número unidad`*`][`*`densidad del medio`*`]fd[`*`sectores por pista`*`]`

donde:

- El número de unidad es 0 para la unidad A, 1 para la B, etc.
- La densidad del medio se especifica en TPIs (Tracks per inch, pistas por pulgada). En la siguiente tabla se listan los valores de este parámetro para formatos típicos de disquetes:

Tipo de disquete	Densidad en TPI
5 ¼", DS-DD	48
5 ¼", DS-HD	96
3 ½", DS-DD y 3 ½", DS-HD	135

- El número de sectores por pista depende también del formato del disco. En la siguiente tabla se pueden ver valores típicos:

Tipo de disquete	Sectores por pista
5 ¼", DS-DD	9
5 ¼", DS-HD	15
3 ½", DS-DD	15
3 ½", DS-HD	18

Por ejemplo, un disquete de 3 ½" de alta densidad insertado en la unidad A se puede acceder con los siguientes nombres de dispositivo:

`/dev/fd0135ds18` (modalidad bloque)
`/dev/rfd0135ds18` (modalidad raw)

En Solaris, el nombre del dispositivo que se usa para acceder a la unidad de disquete cambia dependiendo de que el disquete esté sin formato, formateado pero sin etiqueta interna o formateado con etiqueta interna. Para averiguar el nombre del dispositivo, se inserta el disquete en la unidad y se ejecuta el comando `eject -d`.

Unidades de cinta
Como las cintas son un medio de grabación secuencial, este tipo de dispositivos sólo de puede acceder en modalidad raw.

En SCO UNIX, las primera unidad de cinta del sistema se accede como `/dev/rct0`, la segunda como `/dev/rct1` y así sucesivamente. Esta nomenclatura se emplea para unidades de cinta de cartucho grande y para todas las que usan tecnología SCSI. Para unidades de mini-cartucho de 40 MB, el nombre de dispositivo es `/dev/rctmini`.

Cuando se accede a las unidades de cinta mediante estos dispositivos, la cinta se rebobina de manera automática cuando el comando termina. Sin embargo, a veces es necesario que la cinta no se rebobine (para posicionarla, por ejemplo). En este caso, las cintas se acceden como `/dev/nrct0`, `/dev/nrct1`, `/dev/nrctmini` y así sucesivamente. Estos nombres de dispositivo se conocen como *dispositivos de no rebobinado* (no-rewind devices).

En Solaris, las unidades de cinta se acceden en modalidad de rebobinado como `/dev/rmt/0` y `/dev/rmt/1` y en modalidad de no rebobinado como `/dev/rmt/0n` y `/dev/rmt/1n`.

Unidades de CD-ROM
En SCO UNIX se emplea el nombre de dispositivo `/dev/cd0` para la primera unidad de CD-ROM y `/dev/cd1` para la segunda. Sus respectivos dispositivos para acceso en modalidad raw son `/dev/rcd0` y `/dev/rcd1`.

En Solaris, el nombre del dispositivo de la unidad de CD-ROM cambia según el CD-ROM que se inserte, al igual que con los disquetes. Puede emplearse el comando `eject -d` para observar el nombre de dispositivo asignado, después de insertar el CD-ROM en la unidad.

ADMINISTRACION DE FILESYSTEMS
A continuación se estudian las labores de administración comunes de los filesystems, como su respaldo, creación, reparación, montaje, desmontaje, verificación de espacio libre y manejo de cuotas de espacio.

COPIAS DE SEGURIDAD

Este es uno de los procesos más descuidados, y sin embargo, uno de los más importantes en la administración diaria del sistema operativo UNIX. Nadie está libre de sufrir pérdida de datos por falla del hardware, manos criminales o catástrofes naturales; y en estos casos las copias de seguridad son la única manera de tener el sistema detenido por el menor tiempo posible. En este punto cabe hacer dos preguntas de capital importancia:

¿Por qué son importantes las copias de seguridad?

Las copias de seguridad son un respaldo de uno de los activos más importantes de toda empresa que haya ingresado a la era de la sistematización: La INFORMACION. Recuérdese que la información correcta ayuda a un buen proceso de toma de decisiones, lo cual se puede ahorrar tiempo y costos, y llevar a la empresa a obtener una ventaja competitiva sostenible.

¿Quién es el responsable de realizar las copias de seguridad dentro de la organización?

Esta responsabilidad va en función de la complejidad del sistema. Lo ideal sería que el departamento de sistemas de cada empresa oficiara como coordinador de la labor, y que cada departamento hiciera copia de su información; sin embargo, esto no tiene sentido si la empresa o el sistema de información son relativamente pequeños. Lo importante es tener un proceso bien definido y documentado.

Esquemas para planeación de copias de seguridad

Existen muchos esquemas para planeación de copias de seguridad (es decir, la periodicidad de las copias y los datos de los cuales se debe hacer copia). A continuación se estudiarán los dos más comunes.

a) **Esquema abuelo - padre - hijo:**

Este esquema funciona de la siguiente manera:

- La primera vez se hace una copia de seguridad de todos los archivos en un juego nuevo de cintas. Este juego se etiqueta con el número 1.
- La segunda vez se hace copia de todos los archivos en otro juego nuevo de cintas. Este otro juego se etiqueta con el número 2.
- La tercera vez se hace copia de todos los archivos en otro nuevo juego, que será el juego número 3.
- La cuarta vez, se hace copia de todos los archivos, pero empleando el juego de cintas #1.
- La quinta vez se emplearía el juego #2, y así sucesivamente se hace rotación de los tres juegos de cintas.

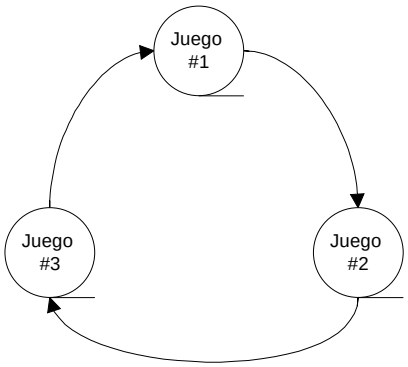


FIGURA 1. Esquema Abuelo-Padre-Hijo para copias de seguridad

El esquema recibe su nombre porque siempre se poseen tres copias de los datos: la más reciente (juego hijo), la anterior al hijo (juego padre) y la anterior al padre (juego abuelo).

Ventajas de este método:

- En caso de desastre la recuperación es rápida, pues solamente debe restaurarse el último juego de cintas.
- Es muy sencillo de comprender e implementar.

Desventajas:

- Exige tener la máquina un gran tiempo fuera de servicio, porque cada proceso de copia es largo.
- Dependiendo de la capacidad de los discos y las cintas, los juegos de cintas tienden a ser grandes.

Para la implementación exitosa de **cualquier** sistema de copias de seguridad, se debe llevar una **bitácora o registro** de las copias de seguridad sonde se especifiquen el grupo de cintas usado, la fecha, hora y nombre de la persona que efectuó la copia. Es muy importante también tener bien definida la periodicidad con la que se harán las copias de seguridad.

b) Esquema incremental

Este esquema funciona de la siguiente manera:

- La primera vez se hace una copia de respaldo total en un juego de cintas nuevo. Este juego de cintas se conoce como copia de seguridad nivel cero.
- La segunda vez se hace copia en un nuevo juego de cintas de los archivos que han sido creados o modificados desde la última copia nivel cero. Este juego de cintas será la copia de seguridad nivel uno.
- La tercera vez se copian solamente los archivos que han sido creados o cambiados desde la última copia nivel uno. Esta será la copia nivel dos.
- Se puede seguir así (empleando un solo juego de cintas por nivel) hasta completar los niveles deseados (muchas aplicaciones de copia de seguridad manejan niveles del 0 al 9; en la práctica se usan generalmente los cuatro primeros).

El siguiente es un ejemplo de esquema de copia incremental que se podría seguir: Hacer copia nivel 0 cada año, copia nivel 1 mensualmente, copia nivel 2 semanalmente y copia nivel 3 diariamente.

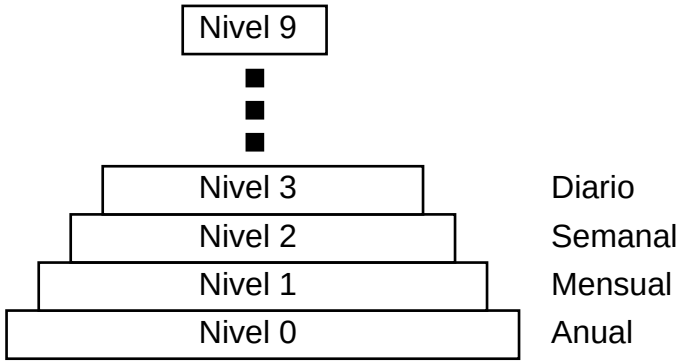


FIGURA 2. Esquema incremental para copias de seguridad

Ventajas de este esquema:

- Las copias completas (más largas) se hacen con menos periodicidad; las más periódicas son muy cortas.
- Se manipulan menos cintas mientras más alto es el nivel de copia.

Desventajas:

- En caso de desastre o pérdida de datos, deben restaurarse TODOS los niveles, comenzando con el 0 y procediendo en orden ascendente.

El corazón de este sistema es un archivo en el cual se almacena la fecha de la última copia de seguridad para cada nivel. La utilidad de backup lo consulta, y de acuerdo a los datos que contiene y al nivel a realizar, selecciona los archivos a copiar.

Debe notarse que para hacer una copia nivel uno o superior, debe existir la copia nivel cero. Por ejemplo, si se desea hacer una copia nivel 2 y no existiera la copia nivel 1, se toma entonces la fecha de la última copia nivel 0. Esto aplica para todos los niveles de copia.

COMANDOS PARA EFECTUAR COPIAS DE SEGURIDAD

Los comandos a estudiar en esta sección son:

- `tar` y `cpio` (disponibles en todas las versiones de UNIX)
- `ufsdump` y `ufsrestore` (disponibles sólo en Solaris)
- `backup` y `restore` (disponibles sólo en SCO UNIX)

`tar`: Tape Archive Program

El comando `tar` genera una copia de respaldo de archivos y/o directorios en un dispositivo magnético, sin generar un catálogo o directorio en el dispositivo.

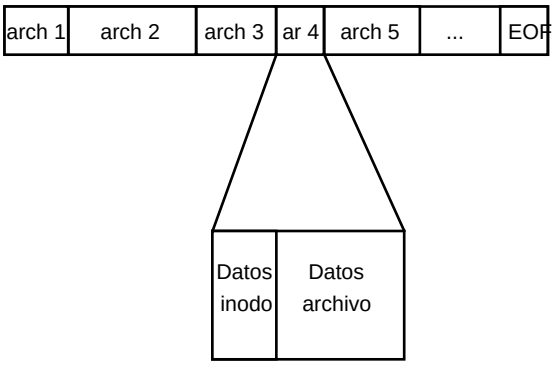


FIGURA 3. Estructura de un archivo `tar`.

Como se puede apreciar en la figura 3, un archivo `tar` consiste en una grabación secuencial de todos los archivos que se especificaron al crear la copia de seguridad. Cada archivo se almacena en dos segmentos; el primero contiene los datos de su inodo y el segundo, el archivo en sí.

El comando `tar` posee tres grandes desventajas:

- No almacena un catálogo de los archivos que graba. Para poder obtener un listado del directorio de la cinta o archivo de copia, hay que recorrerlo en su totalidad.
- No detecta defectos en el medio de grabación.
- Muchas versiones de UNIX no permiten que un respaldo hecho con `tar` ocupe más de un disco o cinta.

La sintaxis del comando es la siguiente:

```
tar opciones archivos
```

Las opciones principales del comando son las siguientes:

- `c` Crea un respaldo nuevo
- `t` Lista el contenido de un respaldo existente
- `x` Extrae el contenido del respaldo
- `v` Muestra el resultado de la operación del comando en pantalla
- `f` Permite especificar el nombre del dispositivo o archivo a usar

- Si la lista de archivos a grabar incluye algún directorio, dicho directorio es recorrido recursivamente y todo su contenido (archivos y subdirectorios) es grabado en el respaldo.
- El dispositivo donde se va a grabar o desde donde se va a leer debe ser accedido en modalidad raw.

Ejemplos:

- El siguiente comando almacena todo el contenido del filesystem `/usr` en la primera unidad de cinta de un equipo Solaris:

```
tar cvf /dev/rmt/0 /usr
```

- El siguiente comando almacena todo el contenido del directorio actual en un disco de alta densidad de 3 ½" en un equipo con sistema SCO UNIX:

```
tar cvf /dev/rfd1135ds18 *
```

En lugar de emplear la opción `f` para especificar el dispositivo, muchas versiones de UNIX permiten acceder por medio de un número los dispositivos más conocidos. La equivalencia entre números y dispositivos se encuentra en el archivo `/etc/default/tar`. A continuación se presenta este archivo en su versión de SCO UNIX:

```
#       device                  block   size    tape
#
archive0=/dev/rfd048ds9         18      360     n
archive1=/dev/rfd148ds9         18      360     n
archive2=/dev/rfd096ds15        10      1200    n
archive3=/dev/rfd196ds15        10      1200    n
archive4=/dev/rfd096ds9         18      720     n
# archive4=/dev/rfd0135ds9      18      720     n
# archive5=/dev/rfd1135ds9      18      720     n
archive5=/dev/rfd196ds9         18      720     n
archive6=/dev/rfd0135ds18       18      1440    n
archive7=/dev/rfd1135ds18       18      1440    n
archive8=/dev/rct0              20      0       y
archive9=/dev/rctmini           20      0       y
#
# The default device in the absence of a numeric or "-f device" argument
archive=/dev/rfd096ds15         10      1200    n
```

En conclusión, el ejemplo anterior podría condensarse así:

```
tar cv7 *
```

- El siguiente comando permite verificar el contenido de una cinta:

```
tar tvf /dev/rmt/0 (en Solaris)
    tar tv8 (en SCO UNIX)
```

- Cualquiera de los dos comandos siguientes permite extraer (restaurar) el contenido de un disquete de alta densidad de 5 ¼" en SCO UNIX:

```
tar xvf /dev/rfd096ds15
        tar xv2
```

Un dato de interés es que el nombre de dispositivo especificado después de la opción `f` puede ser el nombre de un archivo normal. Esto posibilita reunir muchos archivos en uno solo, con el fin de transferirlos más fácilmente de un equipo a otro a través de una red. El archivo así obtenido se puede comprimir con la utilidad `gzip` o una similar. En el Internet, la mayor parte del software de dominio público para UNIX se distribuye de esta manera; los archivos con extensión `.tar.gz` y `.tar.Z` que se pueden encontrar en los servidores de ftp anónimo no son otra cosa que paquetes de software reunidos con `tar` y luego comprimidos.

`cpio`: Copy files in/out

El comando `cpio` permite hacer copia de archivos en modo similar a como lo hace `tar`. `cpio` posee dos ventajas sobre `tar`:

- Hace chequeo del medio de grabación.
- Permite que el respaldo ocupe múltiples cintas o discos.

El comando `cpio` posee dos sintaxis, según se vaya a hacer copia o se vayan a restaurar archivos:

`cpio -o`: Efectuando una copia de seguridad

El comando `cpio -o` lee el nombre o los nombres de los archivos a copiar desde stdin, y copia dichos archivos a stdout en un formato especial.

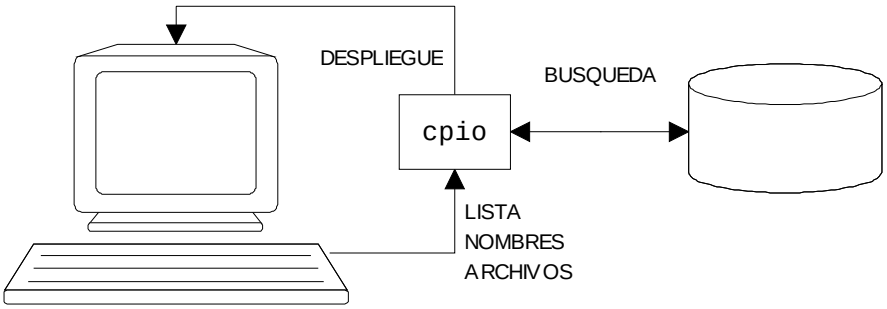


FIGURA 4.
Funcionamiento del comando `cpio -o`

El siguiente comando copia el contenido del directorio home del usuario a la primera unidad de cinta de una máquina con sistema Solaris:

```
ls -R $HOME | cpio -o > /dev/rmt/0
```

La lista de archivos a copiar se introduce a `cpio` por medio de un comando `ls`, y su salida se redirecciona al medio de almacenamiento.

Después de la opción `-o` pueden emplearse las siguientes:

– `c` Escribe el encabezado del archivo `cpio` en ASCII. Por omisión este encabezado se escribe en binario; se puede especificar esta opción para asegurar compatibilidad de la copia entre diferentes versiones de UNIX.
– `v` Muestra mensajes por pantalla a medida que el comando se ejecuta.
– `B` Emplea bloques de 5120 bytes para escribir al medio.

`cpio -i`: Restaurando una copia de seguridad

El comando `cpio -i` lee de stdin un conjunto de datos grabados previamente con el comando `cpio -o`, y extrae de dicho conjunto de archivos aquellos cuyos nombres coincidan con un patrón dado. Como patrón se puede usar cualquier nombre de archivo y se pueden incluir metacaracteres.

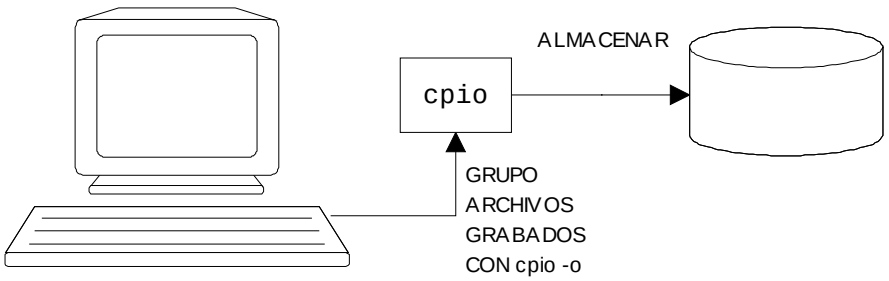


FIGURA 4.
Funcionamiento del comando `cpio -i`

El siguiente comando restaura el archivo `syslog` de una cinta grabada con `cpio` insertada en la primera unidad de cinta de un equipo con SCO UNIX en el directorio actual de trabajo:

```
cpio -i "syslog" < /dev/rct0
```

Nótese el empleo del símbolo de redirección. El patrón o patrones usados deben ser encerrados entre comillas, con el fin de aislar los metacaracteres del shell.

Después de la opción `-i` se pueden emplear las siguientes:
– `d` Crear directorios que no existan.
– `t` Listar solamente el contenido del medio sin restaurar datos.
– `r` Permite renombrar los archivos en forma interactiva.
– `v` Muestra mensajes por pantalla a medida que el comando se ejecuta.

Copias de seguridad en Solaris: `ufsdump` y `ufsrestore`

Solaris dispone de dos programas utilitarios para la gestión de las copias de seguridad; se estudiarán ambos a continuación:

`ufsdump`: Copia de seguridad de un filesystem

El comando `ufsdump` permite hacer copias de seguridad de filesystems completos. Emplea el esquema de copia incremental. La gran ventaja que tiene sobre los sistemas de copia como `cpio` y `tar`, es que crea un catálogo de los archivos que copia, posibilitando así una recuperación más fácil.

La sintaxis del comando es:

```
ufsdump [opciones] filesystem
```

Algunas de las opciones del comando son:

- `nivel` Un número de 0 a 9, que representa el nivel de copia incremental.
- `c` Fija los parámetros para hacer copia en unidad de cartucho.
- `f` Permite especificar el dispositivo en el que se va a almacenar la copia.
- `u` Actualiza la fecha de copia en el archivo `/etc/dumpdates`

Notas:

- Si se desea que el sistema lleve control de qué archivos deben incluirse en la copia según el nivel, debe incluirse la opción `-u` al invocar el comando.
- Para especificar el filesystem se emplea el nombre del dispositivo para acceso en modalidad raw.

El siguiente ejemplo hace copia de seguridad del filesystem root a la primera unidad de cinta de un equipo bajo Solaris:

```
ufsdump 0cfu /dev/rmt/0 /dev/rdsk/c0t3d0s0
```

`ufsrestore`: Restauración de una copia de seguridad

El comando `ufsrestore` permite recuperar copias de seguridad efectuadas con `ufsdump`. Trabaja de manera interactiva y permite navegar con facilidad por el catálogo de la cinta, haciendo muy sencilla la labor de restauración.

La sintaxis del comando es:

```
ufsrestore [opciones]
```

Algunas de las opciones del comando son:

- `i` Habilita la modalidad interactiva de restauración.
- `f` Especifica el dispositivo en el que está almacenada la copia de seguridad.
- `v` Presenta mensajes ampliados cuando se están restaurando los archivos.
- `r` Restaura la cinta completa.

Nota: El comando `ufsrestore` debe invocarse desde el directorio raíz del filesystem a restaurar.

Por ejemplo, para restaurar algunos archivos del filesystem `/home`, se podría dar la siguiente serie de comandos, una vez se cargue la primera cinta del juego adecuado en la primera unidad de cinta del equipo:

```
cd /home
ufsrestore -ivf /dev/rmt/0
```

El comando procede a leer el catálogo de la cinta y luego presenta el prompt `ufsrestore>`, quedando a la espera de subcomandos. Puede usar el subcomando `?` para obtener ayuda al respecto.

Si desea más información acerca del funcionamiento de estos comandos, puede consultar sus respectivos manuales en línea.

Copias de seguridad en SCO UNIX: Comandos backup y restore.

El comando `backup` permite hacer copia de un filesystem completo en un sistema SCO UNIX; `restore` permite hacer una restauración total o parcial de dicha copia de seguridad.

Estos comandos son realmente un front-end del comando `cpio`, es decir, se constituyen en una manera más amigable de emplear este comando. Las cintas generadas con estos comandos están grabadas por lo tanto en formato `cpio` y se puede restaurar con `cpio -i` si hiciere falta.

Los comandos `backup` y `restore` se acceden a través de `sysadmsh`, la utilidad de administración de SCO UNIX basada en menúes.

Menú principal de `sysadmsh`:

```
                                                                    SysAdmSh
System Backups Accounts Printers Media Jobs Dirs/Files Filesystems User Quit
Perform backups of files, filesystems, or the entire system
/u/juancho                                          Monday April 7, 1997 11:28
```

Menú `Backups`:

```
                                                                      Backups
Create Restore Schedule View Integrity
Creates backups
/u/juancho                                          Monday April 7, 1997 11:31
```

Menú ***Create unscheduled:***

```
                                                                    Unscheduled
Press <F3> to choose from a list of filesystems

/u/juancho                                          Monday April 7, 1997 11:32

+-----------------------------Archive Filesystem-------------------------------+
|                                                                              |
|           Filesystem to archive  :   [/dev/root                        ]     |
|                                                                              |
|           Media                  :   [/dev/rct0                        ]     |
|                                                                              |
|           Block size in Bytes    :   [10240   ]                              |
|                                                                              |
|           Volume size in Kbytes  :   [500000  ]                              |
|                                                                              |
|           Format a floppy        :   [Yes]      No                           |
|                                                                              |
|                                                                              |
|           Press <Return> to backup the filesystem or <ESC> to abandon        |
|                                                                              |
|                             [Archive]                                        |
+------------------------------------------------------------------------------+
```

Menú ***Restore Full:***

```
                                                                         Full
Press <F3> to choose from a list of filesystems

/u/juancho                                          Monday April 7, 1997 11:33

+-----------------------------Restore Filesystem-------------------------------+
|                                                                              |
|           Filesystem to restore  :   [/dev/u                           ]     |
|                                                                              |
|           Media                  :   [/dev/rct0                        ]     |
|                                                                              |
|           Block size in Bytes    :   [10240   ]                              |
|                                                                              |
|                                                                              |
|           Press <Return> to restore the filesystem or <ESC> to abandon       |
|                                                                              |
|                                                                              |
|                             [Restore]                                        |
|                                                                              |
|                                                                              |
|                                                                              |
+------------------------------------------------------------------------------+
```

Menú `Restore Partial`*:*

```
                                                                        Partial
Press <F3> to choose from a list of available media

/u/juancho                                          Monday April 7, 1997 11:34

+---------------------------------Restore File---------------------------------+
|                                                                               |
|           Media                   :   [/dev/rct0                        ]     |
|                                                                               |
|           File to restore         :   [usr/initBDIC.ora                  ]    |
|                                                                               |
|           Directory to restore to :   [/                                 ]    |
|                                                                               |
|           Block size in Bytes     :   [10240   ]                              |
|                                                                               |
|                                                                               |
|           Press <Return> to restore the file or <ESC> to abandon              |
|                                                                               |
|                                                                               |
|                              [Restore]                                        |
|                                                                               |
+-------------------------------------------------------------------------------+
```

CREACION DE FILESYSTEMS

El proceso de creación de filesystems se lleva a cabo en las siguientes situaciones:

- Durante el montaje del sistema operativo.
- Cuando se adicionan discos duros al sistema.
- Cuando se desea crear un filesystem en un disquete.

El proceso de creación de un filesystem incluye el chequeo del medio magnético y el aislamiento de sus zonas defectuosas, y posteriormente la creación de las estructuras ya conocidas del bloque de boot, el superbloque, la zona de inodos y la zona de bloques de datos.

Creación de un filesystem en disco duro

Para crear un filesystem en un disco duro se deben verificar los siguientes pasos:

- Instalación del disco duro y reconocimiento del mismo por parte del sistema: En la mayoría de los casos, basta con instalar el disco duro en la máquina para que el sistema lo reconozca. Sin embargo, en algunos casos es necesario instalar una rutina controladora (driver) proveída por el fabricante del disco. En tal caso, deben seguirse con cuidado las indicaciones del fabricante.

- Particionamiento del disco: Como se discutió en un principio, en un disco duro pueden existir uno o varios filesystems. Para informar al sistema operativo cuántos filesystems se desean instalar en el nuevo disco duro, se debe instalar en el disco una

tabla de división. El comando que permite instalarla en SCO UNIX se llama `divvy`, su equivalente en Solaris es el comando `format`.

- Creación de los filesystems: Una vez reservado el espacio, se debe crear la estructura del filesystem.

En SCO UNIX se emplea el comando `mkdev fs` para crear un filesystem. El comando busca entonces divisiones del disco donde todavía no se hayan creado filesystems, y permite al usuario escoger en cuál o cuáles desea crear filesystems. Una vez creados, pregunta si se desea que se monten cada vez que arranque la máquina, y en qué punto.

En Solaris se pueden emplear dos comandos para crear un filesystem, una vez ha sido particionado el disco:

➔ El comando `mkfs` crea un filesystem, pero hay que saber todos los detalles técnicos acerca de la geometría del disco y las características de la controladora, porque el comando exige estos datos como parámetros (puede consultar más detalles en el manual en línea de `mkfs`)

➔ El comando `newfs` actúa como un front-end para `mkfs`. `newfs` averigua todos los detalles técnicos del disco haciendo una consulta al hardware (los datos normalmente se encuentran en memorias ROM de la controladora y del disco), y luego invoca a `mkfs` con los parámetros adecuados. La sintaxis de `newfs` es la siguiente:

```
newfs nombre_dispositivo_bloque
```

Creación de un filesystem en disquete

Los filesystems pueden crearse también en disquetes, para almacenar información como en cualquier otro filesystem, o para emplearlos como discos de arranque para recuperación del sistema en caso de falla, si el computador afectado arranca desde disquete.

En el caso de SCO UNIX, se crean filesystems en disquete con el comando:

```
mkdev fs nombre_dispositivo_raw
```

Por ejemplo: `mkdev fs /dev/rfd096ds15`

El comando da entonces tres opciones:

- Crear únicamente el filesystem.
- Crear el filesystem y un disco de boot, que puede servir para arrancar la máquina.
- Crear el filesystem, un disco de boot y copiar al filesystem los comandos básicos de administración del sistema (de hecho, construir un pequeño filesystem root en el disquete). Este juego de discos sirve para iniciar labores de recuperación en caso de falla del disco duro.

Para crear un filesystem en Solaris, se averigua el nombre del dispositivo del disquete con el comando `eject -d`, y luego se invoca el comando `newfs` con este nombre de dispositivo.

MONTAJE DE FILESYSTEMS

El proceso de montaje encadena una estructura de filesystem con el filesystem raíz, de suerte que el nuevo filesystem se ve como si fuera un subdirectorio dentro del filesystem raíz. El usuario percibe todo esto como una única estructura de almacenamiento.

Acerca del proceso de montaje cabe anotar lo siguiente:

- En el filesystem raíz debe crearse un directorio vacío. Este subdirectorio será el punto de montaje para el nuevo filesystem.
- Si un filesystem no está montado, no se pueden acceder sus datos.

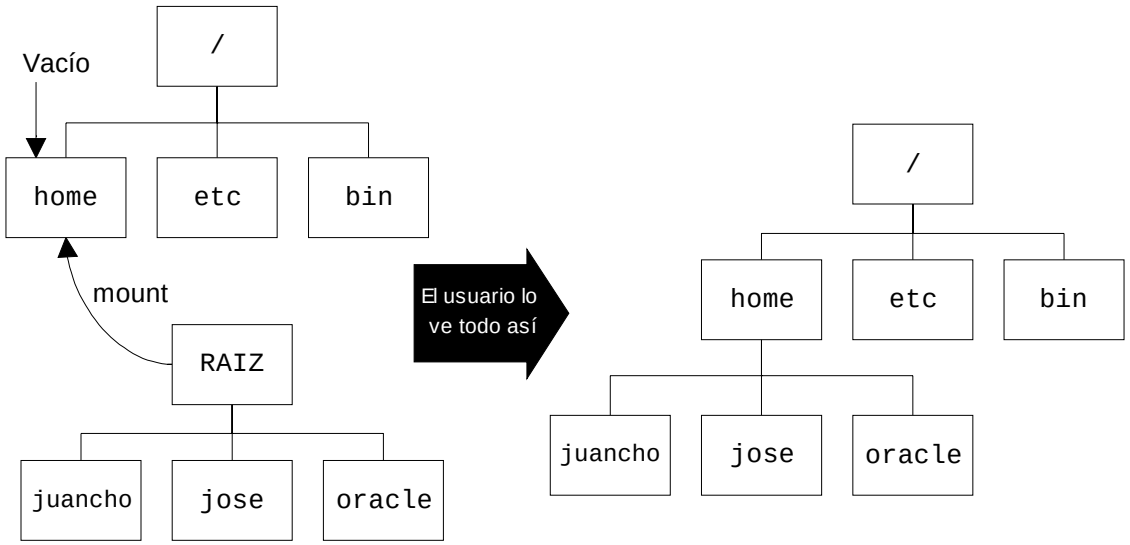


FIGURA 5. El concepto de montaje de un filesystem

El comando `mount` crea la unión entre el filesystem y el directorio vacío del filesystem raíz, y deja el contenido del filesystem a disposición de los usuarios. La sintaxis del comando `mount` es la siguiente:

```
mount [opciones] dispositivo_filesystem punto_de_montaje
```

A continuación se muestran cuatro ejemplos de montaje de filesystems; los dos primeros corresponden a filesystems en disco duro, el tercero a un disquete y el cuarto a un CD-ROM:

```
mount /dev/dsk/c0t3d0s6 /home
mount /dev/u /u
mount /dev/fd1135ds18 /diskette
```

```
mount -r /dev/cd0 /cdrom
```

El nombre del dispositivo que se especifique debe ser el que permite accederlo en modalidad bloque.

Algunas opciones del comando `mount` son las siguientes:
- `r` Monta el filesystem como un dispositivo de sólo lectura.
- `t tipo` Permite especificar el tipo de filesystem. Algunos tipos son `ufs` (Unix Filesystem), `hsfs` ó `iso9660` (para CD-ROMs) y `msdos` (para discos en formato DOS)

No es necesario especificar el tipo en muchos casos, porque el sistema operativo lo detecta de manera automática.

Montaje automático de los filesystems en el arranque del sistema
La mayoría de versiones de UNIX permite especificar cuáles filesystems se deben montar en el momento del arranque del sistema. Por omisión, UNIX monta únicamente el filesystem root, pero el montaje del resto se puede especificar editando un archivo. Este archivo se llama `/etc/vfstab` en Solaris, y `/etc/fstab` en Linux.

El archivo tiene típicamente esta estructura (se muestra la de Solaris):

```
#device                 device                  mount   FS      fsck    mount   mount
#to mount               to fsck                 point   type    pass    at boot options
#
/dev/dsk/c0t3d0s0       /dev/rdsk/c0t3d0s0      /       ufs     1       no      -
/dev/dsk/c0t3d0s6       /dev/rdsk/c0t3d0s6      /usr    ufs     2       no      rq
/dev/dsk/c0t1d0s6       /dev/rdsk/c0t1d0s6      /home   ufs     5       yes     rq
/dev/dsk/c0t1d0s7       /dev/rdsk/c0t1d0s7      /home2  ufs     6       yes     rq
/dev/dsk/c0t3d0s5       /dev/rdsk/c0t3d0s5      /opt    ufs     7       yes     -
```

La especificación de las columnas se da a continuación:
- La primera indica el nombre de dispositivo del filesystem a montar.
- La segunda indica el nombre de dispositivo del filesystem en modalidad de acceso raw (fsck chequea los filesystems en modalidad raw en el caso de Solaris).
- La tercera indica el punto de montaje.
- La cuarta especifica el tipo de filesystem.
- La quinta, el orden en el cual fsck chequeará los filesystems cuando sea necesario.
- La sexta indica si el filesystem debe ser montado o no en el proceso de arranque.
- La última son las opciones de montaje del filesystem. Varían según el tipo del mismo.

DESMONTAJE DE FILESYSTEMS
Con el proceso de desmontaje los datos del filesystem se vuelven inaccesibles al usuario. Seguidamente toda la I/O pendiente a ese filesystem se lleva a cabo (proceso de vaciado de cachés o *cache flushing*) y por último se rompe la conexión entre el filesystem raíz y el que se está desmontando.

Los filesystems se desmontan en los siguientes casos:

- Cuando se desea hacerlos inaccesibles a los usuarios, por ejemplo para efectuar copias de seguridad.
- Cuando se va a apagar el equipo.
- Cuando el medio de almacenamiento es amovible (disquete o CD-ROM) y se va a retirar del equipo.

Es importante desmontar bien los filesystems antes del apagado, para evitar la creación de inconsistencias en su estructura y la posible pérdida de datos.

El comando `umount` ejecuta esta tarea. Su sintaxis es la siguiente:

```
umount dispositivo_filesystem
```

Debe emplearse, al igual que con `mount`, el nombre de dispositivo para acceso en modalidad bloque.

Nota: Si por algún motivo se desea llevar a cabo toda la I/O pendiente en todos los filesystems sin necesidad de desmontarlos, se puede emplear el comando `sync`. Normalmente el sistema operativo ejecuta este comando durante el proceso de apagado antes de desmontar todos los filesystems.

CHEQUEO DE LA INTEGRIDAD DE FILESYSTEMS

Existen casos en los que un filesystem no puede ser desmontado correctamente, como por ejemplo:

- Falla del sistema operativo (bloqueo).
- Falla en la energía.
- Remoción de un medio amovible sin desmontarlo.

En estos casos, la sincronización entre los datos en memoria y los datos en disco (el vaciado de cachés) no se lleva a cabo, y por tanto la estructura del filesystem puede quedar inconsistente. Un filesystem dañado de esta manera no puede ser montado.

En el proceso de arranque, cuando el sistema operativo se da cuenta de que los filesystems fueron incorrectamente desmontados, ejecuta el comando `fsck` sobre cada filesystem dañado y posteriormente procede a montarlos.

El comando `fsck` puede ser ejecutado manualmente en cualquier momento. La única condición para ejecutarlo es que el filesystem debe estar desmontado (la excepción a la regla es el filesystem root).

El proceso ejecutado por `fsck` se divide en 6 fases:

🗐 **FASE 1. Chequeo de bloques y tamaños**

- Se verifica primero la lista de inodos usados.
- Luego se verifica qué bloques de datos han sido encadenados a un inodo, con el fin de detectar bloques duplicados o malos.
- Procede luego a verificar el formato del inodo.
- Por último, chequea el tipo de archivo que le ha sido asignado al inodo.

🗐 **FASE 2. Chequeo de trayectorias y nombres**

- En esta fase se borran las entradas de directorio que apuntan a archivos o directorios modificados en la fase 1.

🗐 **FASE 3. Chequeo de conectividad**

- Se verifica que cada inodo usado aparezca al menos en una tabla de directorio, y que los múltiples links de un archivo tengan sentido.

🗐 **FASE 4: Chequeo del conteo de referencia**

- Esta fase está relacionada con el conteo de links, y con las modificaciones que pudieron haberse efectuado en la fase anterior.
- En esta fase se reportan y se corrigen todos los errores relacionados con: Archivos no referenciados, conteo incorrecto de links, bloques de datos malos o duplicados, y suma incorrecta de inodos libres.

🗐 **FASE 5: Chequeo de la lista de libres**

- Cuando `fsck` calcula el número de bloques libres, y éste no coincide con la información de la lista de bloques libres, reporta en esta fase una condición de error.

🗐 **FASE 6: Grabación de la lista de libres**

- Si la fase 5 reporta una condición de error, la lista de bloques libres es reconstruida y grabada.

El comando `fsck` es eficaz el 99% de las veces. El margen de error existente es indicativo de la importancia de tener una buena política para efectuar copias de seguridad. Si `fsck` falla, los datos contenidos en el filesystem dañado serán completamente inaccesibles.

CONTROL DE ESPACIO LIBRE EN UN FILESYSTEM

Es recomendable operar un equipo UNIX con los filesystems por debajo del 85% de ocupación. Esto con el fin de tener espacio suficiente para mover archivos grandes, y también porque el sistema tiende a volverse inestable cuando los filesystems se llenan (se pueden perder datos).

Control del espacio libre a nivel de filesystems: Comando `df`

El comando `df` permite obtener un reporte de la ocupación de todos los filesystems que se encuentran montados en ese momento. Debe recordarse que un filesystem tiene dos estadísticas de ocupación: Porcentaje de bloques ocupados y porcentaje de inodos ocupados.

El comando `df -k` permite verificar la capacidad en bloques ocupada y libre de los filesystems de un sistema Solaris; `df -v` hace el trabajo equivalente en SCO UNIX.

```
/home/juancho> df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0      76767   49483   19614    72%    /
/dev/dsk/c0t3d0s6     635862  545358   26924    95%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
swap                  183876       4  183872     0%    /tmp
/dev/dsk/c0t1d0s6     481823  179623  254020    41%    /home
/dev/dsk/c0t1d0s7     481823  158939  274704    37%    /home2
/dev/dsk/c0t3d0s5      96455   64936   21879    75%    /opt
```

El comando `df -i` de SCO UNIX (`df -e` en Solaris) permite verificar la capacidad ocupada y disponible de inodos.

Ejemplo de SCO UNIX:

```
160# -/u/juancho> df -i
Mount Dir  Filesystem              iused     ifree    itotal  %iused
/          /dev/root                9306     56182     65488    14%
/u         /dev/u                   3299     62189     65488     5%
```

Ejemplo de Solaris:

```
/home/juancho> df -e
Filesystem              ifree
/dev/dsk/c0t3d0s0       36690
/dev/dsk/c0t3d0s6      299622
/proc                    1199
fd                          0
swap                    15731
/dev/dsk/c0t1d0s6      235950
/dev/dsk/c0t1d0s7      238686
/dev/dsk/c0t3d0s5       47848
```

Consulte el manual en línea del comando para ver opciones específicas a la versión de UNIX que esté empleando.

Control de espacio ocupado por archivos y directorios: Comando du

El comando `du` (Disk Usage) permite verificar la cantidad de espacio en bloques de disco que ocupa un archivo o un directorio con todos los subdirectorios y archivos que dependan de él. La sintaxis del comando es la siguiente:

```
du [opciones] archivo|directorio
```

El comando `du -a` muestra, en el caso de un directorio, el espacio ocupado por todos los subdirectorios y archivos que dependen de él, y finalmente presenta el total:

```
/usr/oracle> du -a
2       ./local.profile
4       ./.profile
2       ./local.cshrc
2       ./local.login
4       ./.sh_history
16      ./.pine-debug1
…
16      ./.pine-debug3
16      ./.pine-debug4
110     .
```

El comando `du -s` muestra solamente la línea de totales final que despliega `du -a`:

```
/home/juancho> du -s
67130   .
```

En estos dos casos, se omitió el parámetro de nombre de archivo. Como se puede ver, du despliega información acerca del directorio actual.

MANEJO DE CUOTAS DE ESPACIO

En algunas ocasiones es necesario controlar el espacio en disco que cada usuario emplea, puesto que se trata de un recurso crítico, y sobre todo en máquinas con muchos usuarios se puede convertir en un problema. Con este fin, cada vez más versiones de UNIX poseen capacidad de manejo de cuotas de espacio por usuario. Se estudiará a continuación el manejo de cuotas en Solaris, que posee esta capacidad.

Habilitación de las cuotas

Las cuotas se habilitan por filesystem. Para habilitar las cuotas en un filesystem, se debe hacer lo siguiente:

- Crear en el directorio raíz del filesystem un archivo llamado `quotas`, con dueño root y permisos 0644 (`-rw-r--r--`).
- Modificar la línea correspondiente al filesystem en el archivo `/etc/vfstab`. En la columna "Mount options" deba cambiarse "`rw`" (read / write) por "`rq`" (read / write with quotas enabled).
- Hechos estos cambios se rearranca el sistema.

Asignación de las cuotas

Una vez efectuado el paso anterior, se procede a fijar la cuota para cada usuario, mediante el comando:

```
edquota nombre_usuario
```

Este comando invoca al editor de texto `vi`, y carga en él una plantilla donde se pueden editar los valores de las cuotas de espacio para el usuario. Existe en dicha plantilla un registro por cada filesystem, y cada registro permite especificar la cuota para dicho

filesystem. Se pueden establecer cuotas de bloques de disco o de inodos, según se ve en el siguiente ejemplo de plantilla:

```
fs /home2 blocks (soft = 100, hard = 110) inodes (soft = 0, hard = 0)
fs /home blocks (soft = 1, hard = 1) inodes (soft = 0, hard = 0)
fs /usr blocks (soft = 1, hard = 1) inodes (soft = 0, hard = 0)
```

Para cada objeto (bloque e inodo) existen además dos tipos de cuota:

- La cuota blanda (soft). Cuando el usuario alcanza este número de objetos utilizados se le alerta de que lo ha alcanzado, pero se le permite usar más objetos.
- La cuota dura (hard). Cuando el usuario alcanza esta número de objetos utilizados se le impide utilizar más.

Para que tengan sentido, la cuota blanda debe ser siempre menor que la cuota dura. Un valor de cero (`0`) en cualquiera de los campos indica que no se establece cuota de uso (se pueden usar todos los objetos que haya disponibles en el filesystem).

Una vez modificados los valores se sale de `vi` grabando el archivo, con lo cual se almacenan los cambios en los archivos `quotas` de los filesystems (estos archivos poseen formato binario y solamente pueden ser editados mediante el comando `edquota`).

Muchas veces se tienen grupos de usuarios con cuotas similares. Para evitar tener que fijar las cuotas individualmente a cada miembro del grupo, se puede usar uno de estos miembros como patrón. Se le fijan entonces las cuotas a este usuario patrón, y después se copian las cuotas al resto de miembros del grupo mediante el comando:

```
edquota -p usuario_patrón usuario [lista de usuarios…]
```

También se pueden fijar cuotas de tiempo, esto es, el tiempo del que disponen los usuarios para borrar archivos cuando alcanzan la cuota dura. Para más detalles acerca de las cuotas de tiempo y el uso del comando `edquota`, puede consultar el manual el línea del comando.

Chequeo de consistencia, encendido y apagado de las cuotas

Una vez creadas las cuotas, se debe chequear la consistencia de la base de datos de cuotas, con el fin de que se registre el uso actual de disco por parte del usuario en dicha base de datos. Este chequeo se debe hacer también cada vez que se modifiquen cuotas. El comando que permite hacerlo es el comando `quotacheck`:

```
quotacheck [-pva] [filesystem]
```

Las opciones del comando tienen el siguiente significado:

– `p` Hace el chequeo de filesystems en paralelo (útil para máquinas con varios procesadores)

- `v` Mostrar mensajes a medida que se realiza la operación
- `a` Chequear todos los filesystems con cuotas habilitadas

Con la base de datos consistente, se pueden encender las cuotas, con lo cual entran en vigor. Esto se hace por medio del comando `quotaon`:

```
quotaon [-va] [filesystem]
```

Las opciones del comando tienen el mismo significado que para `quotacheck`.

Si en determinado momento se deseara inhabilitar las cuotas temporalmente, se puede hacer por medio del comando `quotaoff`:

```
quotaoff [-va] [filesystem]
```

Las opciones del comando tienen también el mismo significado que para `quotacheck`.

Debe anotarse que Solaris ejecuta durante el proceso de arranque los comandos `quotacheck` y `quotaon`, para de esta manera asegurar la consistencia de la base de datos y la actividad del sistema de cuotas en cada arranque de la máquina.

Elaboración de reportes de cuotas

Si se desea saber cómo están haciendo uso los usuarios del espacio asignado por el sistema de cuotas, se puede obtener un reporte para tal fin. El comando que se usa es `repquota`:

```
repquota [-va] [filesystem]
```

Las opciones tienen el mismo significado que para el comando `quotacheck`. A continuación se muestra un fragmento de un reporte típico obtenido con este comando:

```
/dev/dsk/c0t3d0s6 (/usr):
                     Block limits                        File limits
User              used    soft    hard     timeleft     used    soft    hard
timeleft
ogrupo0   --      0      1      1                  0      0      0
ogrupo1   --      0      1      1                  0      0      0
ogrupo2   --      0      1      1                  0      0      0
ogrupo3   --      0      1      1                  0      0      0
ogrupo4   --      0      1      1                  0      0      0
ogrupo5   --      0      1      1                  0      0      0
ogrupo6   --      0      1      1                  0      0      0
```

Los usuarios individuales pueden obtener un reporte del uso de su propia cuota ejecutando el comando `quota -v`. A continuación se presenta un ejemplo del reporte que genera este comando:

```
Disk quotas for gritter (uid 283):
Filesystem     usage  quota  limit    timeleft  files  quota  limit    timeleft
/usr               0      1      1                   0      0      0
/home              0      1      1                   0      0      0
/home2            68    100    110                  19      0      0
```

MANEJO DE MEDIOS DE ALMACENAMIENTO CON FORMATO DOS

Muchas versiones de UNIX permiten leer y escribir medios magnéticos con formato DOS. El procedimiento varía según la versión del sistema; se estudiarán los casos de SCO UNIX y Solaris.

SCO UNIX

Esta versión de UNIX posee una serie de comandos compatibles con DOS. Los medios magnéticos se referencian con la nomenclatura usual de DOS (`a:`, `b:`, `c:`, etc.).

Los comandos son:

- `dosdir`: Muestra el directorio del disco.
- `doscp`: Copia archivos de un disco DOS a un filesystem UNIX, y viceversa.
- `dosrm`: Borra un archivo.
- `dosmv`: Cambia el nombre a un archivo.
- `doscat`: Despliega el contenido de un archivo.
- `dosformat`: Formatea un disco en formato DOS.

Para mayores detalles, consultar el manual en línea de estos comandos.

SOLARIS

Solaris posee un proceso llamado `vold` (Volume Management Daemon), que se encarga de gestionar todos los medios de almacenamiento amovibles tipo disco, como son los disquetes y los CD-ROMs.

Si se desea emplear un disquete con formato DOS, basta con introducirlo en la unidad y ejecutar el comando `volcheck` para avisar al proceso `vold` de la inserción del disquete. `vold` lo montará entonces en el subdirectorio `/floppy/floppy0` como si se tratara de un filesystem, y se podrá interactuar con su contenido como se hace con los archivos normales de UNIX. Una vez se termine de usar el disquete, se debe ejecutar el comando `eject` para desmontarlo y expulsarlo.

Para la lectura de CD-ROMs, el proceso es similar. Las únicas diferencias son las siguientes:

- No es necesario ejecutar el comando `volcheck` al insertar el CD-ROM, porque la unidad es capaz de alertar al proceso por sí sola.
- El CD-ROM queda montado en el directorio `/cdrom/cdrom0`.