



Preparación para el examen LPI 101

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

Créditos y licencia de uso

Coordinación:

Manuel Guillán (xLekOx) lpi@xlekox.org

Oscar Casal (ocs) oscar@glug.es

Traducción:

Pere Catalan (arGos) sageta77@hotmail.com

Juan Maria Gil (Smooth) yo@juanmaria.com

Ivan Servia (katas) ivanservia@hotmail.com

Pablo Taboada (java) ptaboada@wanadoo.es

Kiefer Von Jammo (Kiefer) kiefer@khrooon.net

Maquetación:

Manuel Guillán (xLekOx) lpi@xlekox.org

Kiefer Von Jammo (Kiefer) kiefer@khrooon.net

Javier Pulido (jpulido) javier.pulido@wanadoo.es

Versión 1.1 (20-09-2004 22:46)

Distribuido por FreeUOC (www.freeuoc.org) bajo licencia: Attribution-NonCommercial-ShareAlike2.0 de commons creative



<http://creativecommons.org/licenses/by-nc-sa/2.0/>

ÍNDICE

Índice de contenido

Tema 104	
Dispositivos, sistemas de ficheros y sus estándares.....	1
Créditos y licencia de uso.....	2
ÍNDICE.....	3
Tema 104.1	
Creando particiones y sistemas de ficheros.....	5
Introducción.....	6
Creando particiones y sistemas de ficheros.....	7
Fdisk.....	7
mkfs.....	10
fips.....	11
Tema 104.2	
Creando particiones y sistemas de ficheros.....	12
Introducción.....	13
Manteniendo la Integridad de los Filesystems (Sistemas de Ficheros).....	14
Monitorizando el espacio y los inodos libres del disco.....	14
Monitorizando el Uso del Espacio en Disco.....	16
Comprobando la Integridad del Filesystem.....	17
Creando un sistema de ficheros.....	21
Algunas utilidades de los filesystems.....	22
Tema 104.3	
Montando y desmontando filesystems.....	23
Introducción.....	24
Controlando el montaje / desmontaje de los Filesystems.....	25
Administrando la tabla de filesystems.....	25
Montando filesystems.....	26
Desmontando filesystems.....	28
Tema 104.4	
Administrando cuotas de disco.....	29
Introducción.....	30
Establecer y Consultar Cuotas de Disco.....	31
Cuotas de Limitación.....	31
Comandos de cuotas.....	32
quota.....	32
quotaon.....	33
quotaoff.....	34
quotacheck.....	35
edquota.....	36

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

repquota.....	37
Habilitando las cuotas.....	38
Tema 104.5	
Usando permisos para controlar el acceso a los ficheros.....	40
Introducción.....	41
Permisos de Archivos y Directorios.....	42
Permisos estándar.....	42
Cambiando Valores.....	45
Permisos especiales.....	46
SUID.....	46
SGID.....	46
Sticky Bit.....	47
Tema 104.6	
Administrando el propietario de los ficheros.....	48
Introducción.....	49
Permisos de Usuarios y Grupos.....	50
chown.....	50
chgrp.....	51
Tema 104.7	
Crear y cambiar enlaces a ficheros.....	52
Introducción.....	53
Creando links a ficheros.....	54
Tema 104.8	
Búsqueda de ficheros.....	55
Introducción.....	56
Buscando Archivos.....	57
find.....	57
locate.....	57
which.....	58
whereis.....	58
PREGUNTAS TEST.....	59
EJERCICIOS DE LABORATORIO.....	60
RESPUESTAS TEST.....	61
RESPUESTAS DE LABORATORIO.....	61
Bibliografía y enlaces recomendados.....	63

Tema 104.1

Creando

particiones y

sistemas de

ficheros

Introducción

En este tema se verá como crear y formatear particiones con las herramientas que ofrece GNU/Linux.

Los comandos que se verán en este tema son:

fdisk
mkfs

En este capítulo no se harán ejercicios, ya que hay suficientes ejemplos para trabajar con el tema y crear o modificar los existentes para hacer pruebas.

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Creando particiones y sistemas de ficheros

Hay muchas herramientas para crear particiones y sistemas de archivos en sistemas Linux. La utilidad `fdisk` se usa para trabajar con particiones en discos duros. La utilidad `fips` se utiliza para redimensionar particiones, y los sistemas de archivos se crean utilizando la utilidad `mkfs`. Se pueden utilizar estas herramientas conjuntamente para preparar un disco duro para usar sistema Linux.

En el mundo real: muchas distribuciones Linux incluyen herramientas que crean automáticamente las particiones y el sistema de archivos Linux durante la instalación. Mucha gente no necesitará nunca usar utilidades como `fdisk` o `mkfs`. Sin embargo, los usuarios avanzados y los administradores de sistemas deben estar familiarizados con el uso de estas herramientas.

Fdisk



La herramienta principal usada para crear particiones de disco es `fdisk`. La utilidad `fdisk` divide el disco en particiones y escribe la tabla de particiones en el sector 0 (conocido como superbloque). Cuando se utiliza sin parámetros, `fdisk` presenta un menú de opciones con las que se puede interactuar. Se puede evitar el menú y ejecutar `fdisk` con las siguientes opciones:

- l: lista las tablas de particiones.
- v: Nos da únicamente la versión de `fdisk`.

Si no se utiliza ninguna de esas opciones, `fdisk` comprueba si el número de cilindros del dispositivo por defecto (`hda1`) es mayor de 1024 y avisa de ello si es así. Entonces espera una instrucción. Se puede iniciar `fdisk` con un dispositivo distinto al de defecto especificándolo en la línea de comandos. Por ejemplo, para arrancar `fdisk` con el tercer driver IDE, se debe poner:

```
$ fdisk /dev/hdc
```

Una vez la utilidad está iniciada, entrando `m` se presenta una ayuda en forma de menú, algunas de las cuales se listan en la tabla 1-1

Tabla 1-1 Algunas de las opciones del `fdisk`

Opción	Función
p	Muestra información sobre la partición.
d	Borra una partición.
n	Crea una partición.
q	Sale de la aplicación sin guardar los cambios.
w	Guarda los cambios y sale de la aplicación.
m	Muestra los comandos disponibles.
v	Verifica la tabla de particiones.
a	Cambia el indicador de estado de arranque de la partición.

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

Los siguientes ejemplos ilustran que se puede hacer con esta utilidad, empezando por acciones simples y acabando con algunas más complicadas.

En primer lugar, para ver la tabla de particiones, se debe introducir el comando `p`. El resultado se debe parecer a:

```
Disk /dev/hda: 16 head, 63 sectors, 16383 cylinders
Units = cylinders of 1008 + 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	+	1	4063	2047720+	83	Linux
/dev/hda2		4064	4316	127512	82	Linux swap
/dev/hda3		4317	16383	6081768	83	Linux

Se muestra tres particiones (de 1 a 3) en un único disco IDE (hda). La primera partición es de arranque (marcada como boot), la segunda de swap, y la tercera es el resto del disco.

Para modificar el sistema de ficheros, vamos a cambiar la tercera partición en dos de menor tamaño. Primero entramos el comando `d` para borrar la partición. El prompt preguntará que partición (1 a 4). Se entra 3, y ya está.

Para crear una nueva partición, se entrará el comando `n`. El prompt que muestra será:

```
e-   extendida
p-   partición primaria (1-4)
```

Si ya se han creado algunas particiones, el prompt puede no aparecer. Por ejemplo, si ya se tiene una partición extendida, no se puede crear otra ya que ésta ya existe. Si se entra `p` para una primaria, seguidamente se pedirá el número (1-4) para crearla; si se da un número ya usado, el comando falla ya que primero se debería haber borrado la partición antes de volverla a añadir.

En este caso, suponiendo que se quiere crear dos particiones iguales en el espacio que antes ocupaba la tercera partición, por lo tanto después de entrar `p` la secuencia será la siguiente (valores entrados en cursiva)

```
Número de particion (1-4): 3
Primer cilindro (4317-16383, defecto 4317): (aceptar)
Usando el valor de defecto 4317
Último cilindro o +size o +sizeM o +sizeK _
(4317-16383, defecto 16383): 10350
```

```
Command (m para ayuda): p
```

```
Disco /dev/had: 16 head, 63 sectors, 16383 cylinders
Units = cylinders of 1008 + 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	+	1	4063	2047720+	83	Linux

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

```
/dev/hda2          4064 4316 127512      82   Linux swap
/dev/hda3          4317 10350 3041136      83   Linux
```

Command (m para ayuda): *n*

Command action

e extended

p primary partition (1-4)

p

Número de partición (1-4): *4*

Primer cilindro (10531-16383, default 10351): (*aceptar*)

Usando el valor de defecto 10351

Último cilindro o +size o +sizeM o +sizeK _

(10351-16383, defecto 16383): (*aceptar*)

Usando el valor de defecto 16383

Command (m para ayuda): *p*

Disco /dev/had: 16 head, 63 sectors, 16383 cylinders

Units = cylinders of 1008 + 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	+	1	4063	2047720+	83	Linux
/dev/hda2		4064	4316	127512	82	Linux swap
/dev/hda3		4317	10350	3041136	83	Linux
/dev/hda4		10351	16383	3040632	83	Linux

Para cambiar la tercera partición a sistema de ficheros swap (partición de swap) la secuencia es:

Command (m para ayuda): *t*

Número de partición (1-4): *3*

Hex code (L para la lista de códigos): *82*

Cambiado tipo de sistema de la partición 3 a 82 (Linux swap)

Entonces, para cambiar la partición existente de swap (la 2) a una partición Linux, se debe hacer lo mismo que se ha hecho, pero como código hex el 83. Poniendo L cuando se pida el hex code se mostrarán todos los sistemas de ficheros posibles (igual que poniendo I en el menú principal de fdisk).

Después de efectuar todos los cambios, se puede salir de fdisk y formatear las particiones que lo necesiten. Si se guardan los cambios, aparecerá una alerta indicando que la tabla de particiones ha sido alterada y el disco será sincronizado. Se deberá reiniciar el sistema para asegurarse de que la tabla está actualizada adecuadamente.

Se pueden hacer muchos cambios con fdisk y deshecharlos si se usa la opción q. Si se desea grabar los cambios realizados, se deben salvar con w.

mkfs



Una vez se ha creado la partición, el sistema de ficheros debe ser añadido para que Linux pueda hacer uso de este espacio. La utilidad mkfs se usa para crear sistemas de ficheros en particiones vacías. La utilidad mkfs se utiliza con muchas opciones distintas, las cuales se pueden ver en la tabla 1-2.

En la tabla 1-3 se pueden ver algunas de las utilidades prácticas de mkfs.

Tabla 1-2 Opciones usadas con mkfs

<i>Opción</i>	<i>Uso</i>
-t fstype	Especifica el tipo de sistema de ficheros a crear. Por defecto se usa ext2.
fs -options	Opciones específicas de sistema de ficheros para ser pasados al sistema real de ficheros que vamos a crear.
-c	Comprueba el dispositivo en busca de bloques defectuosos antes de crear el sistema de ficheros.
-l fichero	Lee los bloques defectuosos del fichero.
-v	Produce una salida con más información, incluyendo todas las órdenes específicas del sistema de ficheros concreto que se ejecutan. Ésto es realmente sólo útil para comprobaciones.

Las opciones usadas por mkfs están seguidas por un argumento especificando la partición que debe ser formateada. Después de la ejecución del comando, el código de salida de 0 (cero) indicará que se ha llevado a cabo con éxito mientras que el código de salida 1 (uno) indicará fallo.

Ejemplo de la sintaxis usada:

mkfs -opciones argumentos

Cuando se crea un sistema de ficheros con mkfs, se pueden utilizar muchos comandos para especificar el tipo de sistema de ficheros. Estos comandos son accesibles sólo para el usuario root.



EXAMEN: Las utilidades a entender para el test, y para el uso en el mundo real, están cubiertas aquí. Cada sistema de ficheros se crea usando un comando específico. Se ha de estar seguro de conocer los comandos usados para crear sistemas de ficheros ext2, swap y MS-DOS.

Tabla 1-3 Utilidades para la creación de Sistemas de Ficheros

<i>Comando</i>	<i>Uso</i>
mkfs.ext2 o mke2fs	Crea un sistema de ficheros ext2
mkfs.msdos o mkdosfs	Crea un sistema de ficheros MS-DOS
mkswap	Crea un sistema de ficheros de Linux swap
mkraid	Inicializa y actualiza cadenas de dispositivos RAID
mkfs.minix	Crea un sistema de ficheros Minix
mkfs.bfs	Crea un sistema de ficheros SCO BFS

Para obtener más detalles de las opciones al crear cada tipo de sistemas de ficheros, se puede hacer uso de las páginas man de Linux.

Para crear un sistema de ficheros debe ser usada la herramienta correcta especificada en la tabla 1-3. Por ejemplo, un uso apropiado de estas utilidades es crear un partición ext2 usando mkfs.ext2:

```
# mke2fs /dev/hda3
```

fips

La utilidad fips no es una utilidad Linux. Esta utilidad no es materia de examen, pero es útil conocerla cuando queremos añadir Linux a un sistema ya existente. Se trata de una utilidad MS-DOS proporcionada con Red-Hat para redimensionar particiones en un disco duro. La utilidad produce dos particiones, la partición original redimensionada y una nueva partición creada a partir del espacio libre. Entonces se necesita borrar la nueva partición creada de modo que el espacio pueda ser utilizado por Linux. Antes de utilizar la utilidad fips es importante hacer copia de seguridad de los datos almacenados en la partición sobre la que se trabajará.

La utilidad se encuentra en el CD-ROM de Red-Hat, en el directorio dosutils. La utilidad se ejecuta desde la línea de comandos de MS-DOS, y los cursores se utilizan para redimensionar las particiones. Se debe tener extremo cuidado utilizando esta utilidad.

En el mundo real: La utilidad fips es similar al PartitionMagic y otros programas comerciales que se utilizan para crear y redimensionar particiones.

Tema 104.2

Creando

particiones y

sistemas de

ficheros

Introducción

Este capítulo se verá como verificar la integridad del disco duro, monitorizar el espacio libre y reparar problemas con el sistema de ficheros.

Los comandos que se verán en este tema son:

- du
- df
- fsck
- e2fsck
- mke2fs
- debugfs
- dumpe2fs
- tune2fs

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Manteniendo la Integridad de los Filesystems (Sistemas de Ficheros)

Con el transcurso del tiempo los filesystems activos pueden terminar presentando problemas como los siguientes:

- Un filesystem se llena hasta el límite de su capacidad, causando que los programas o, quizás, el sistema entero dejen de funcionar.
- Un filesystem se corrompe, seguramente por un corte de alimentación o por una caída del sistema.
- Un filesystem se queda sin inodos libres de tal forma que no se pueden crear nuevos objetos en el mismo.

Monitorizar y comprobar cuidadosa y regularmente los filesystems de Linux nos ayudará a prevenir o corregir estos problemas.

Monitorizando el espacio y los inodos libres del disco

Un sistema de lectura/escritura no sirve de mucho si crece hasta el punto en que no pueda admitir nuevos ficheros. Esto podría ocurrir si el filesystem se llena o si se queda sin inodos libres. Los inodos son las estructuras de datos dentro del filesystem que describen los ficheros en el disco. Cada filesystem contiene un número finito de inodos que se establece en el momento de creación del filesystem. Este número es, a su vez, el máximo número de ficheros que un filesystem puede acomodar. Como los filesystems se crean con un número de inodos enorme, probablemente nunca crearás tantos ficheros como para agotar este número. No obstante, es posible quedarse sin inodos libres en particiones que contengan muchos ficheros pequeños.



Es muy importante prevenir la escasez de inodos libres en las particiones del sistema. El comando *df* proporciona información necesaria tanto sobre la uso del espacio en disco como de los inodos libres.

Sintaxis

```
df [opciones] [directorios]
```

Descripción

Muestra información general sobre el uso del disco en los filesystems montados en *directorios*. Normalmente, en *directorios* indicamos ficheros de dispositivos de particiones como */dev/hda1*, pero si indicamos otro tipo de nombre de fichero o directorio obtendremos información sobre la partición donde está ubicado dicho fichero o directorio. Si omitimos *directorios*, se mostrará la información relativa a los filesystems montados en los dispositivos incluidos en */etc/fstab*.

Tabla 2-1 Opciones frecuentemente del comando df

Opción	Uso
-h	Muestra los resultados en un formato legible para las personas, incluyendo sufijos como M(megabytes) y G (gigabytes).
-i	Muestra información sobre los inodos libres en lugar de la información por defecto sobre el espacio libre en disco.

Ejemplo 1

Revisar el uso del espacio en disco en todos los filesystems:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       387M  56M  311M  15% /
/dev/sda5       296M  5.2M  276M   2% /boot
/dev/sda9       1.9G  406M  1.4G  22% /home
/dev/sda6       53M   12M   39M  23% /root
/dev/sda10      99M  104k   93M   0% /tmp
/dev/sda8       972M  507M  414M  55% /usr
/dev/sda7       296M   9.3M  272M   3% /var
```

En este ejemplo se observa que en ninguno de los siete filesystems montados por defecto, el espacio utilizado excede del 55 por ciento de su capacidad.

Ejemplo 2

Revisar el uso de inodos en los mismos filesystems:

```
# df -i
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/sda1       102800  7062  95738   7% /
/dev/sda5       78312   29  78283   0% /boot
/dev/sda9       514000   934  513066   0% /home
/dev/sda6       14056   641  13415   5% /root
/dev/sda10      26104   60  26044   0% /tmp
/dev/sda8       257040  36700  220340  14% /usr
/dev/sda7       78312   269  78043   0% /var
```

Entre estas particiones el mayor consumo de inodos es sólo de un 14 por ciento. Está claro que ninguno de éstos filesystems se está acercando en su consumo al máximo disponible. Observa que la partición */usr* (con el 14 por ciento de los inodos utilizados) ha consumido el 55 por ciento del espacio en disco. Con ésta tónica de utilización, lo más probable es que el volumen */usr* agote su capacidad en disco antes de agotar los inodos libres.

Ejemplo 3

Determina rápidamente en que partición está situado el directorio de trabajo actual (puede representarse, simplemente, por un punto):

```
# df .
/dev/sda1       102800  7062  95738   7% /
```

Cuando un filesystem está próximo a agotar su capacidad podemos, simplemente, eliminar ficheros para obtener más espacio libre. Sin embargo en el caso improbable de una escasez de inodos, deberíamos volver a crear el filesystem con un número mayor de inodos a menos que podamos borrar una muy buena cantidad de ficheros.

Monitorizando el Uso del Espacio en Disco



¿Te has preguntado alguna vez “A donde va todo el espacio consumido en el disco?”. En algunos sistemas operativos es bastante complicado obtener la respuesta a esta pregunta con las herramientas nativas. En Linux, el comando *du* nos puede ayudar, mostrándonos directorio por directorio el uso del espacio en disco, a responder a esta pregunta. El comando *du* examina los directorios recursivamente y muestra información detallada o resumida sobre el espacio en disco consumido.

Sintaxis

```
du [opciones] [directorios]
```

Descripción

Muestra información sobre el uso del disco en los *directorios*. Si se omiten los *directorios* se buscará en el directorio de trabajo actual.

Tabla 2-2 Opciones frecuentemente utilizadas por du

Opción	Uso
-a	Muestra todos los ficheros, no solo los directorios.
-c	Genera un gran total de todos los elementos listados.
-h	Muestra los resultados en un formato legible para las personas, incluyendo sufijos como M (megabytes) y G (gigabytes).
-s	Visualiza un sumario para cada uno de los <i>directorios</i> especificados, en lugar de los totales encontrados recursivamente en cada subdirectorio.
-S	Excluye los subdirectorios de las sumas y los totales, limitándose a totalizar los <i>directorios</i> .

Ejemplo 1

Examinar el uso del disco en */etc/rc.d*:

```
# du /etc/rc.d
882  /etc/rc.d/init.d
1    /etc/rc.d/rc0.d
1    /etc/rc.d/rc1.d
1    /etc/rc.d/rc2.d
1    /etc/rc.d/rc3.d
1    /etc/rc.d/rc4.d
1    /etc/rc.d/rc5.d
1    /etc/rc.d/rc6.d
904  /etc/rc.d
```

Ejemplo 2

Muestra el espacio en disco utilizado por ficheros, incluyendo los subdirectorios interiores, en */etc*:

```
# du -s /etc
13002 /etc
```


Ejemplo 3

Muestra el espacio en disco utilizado por ficheros, excluyendo los subdirectorios interiores, en */etc*:

```
# du -Ss /etc
1732 /etc
```

Ejemplo 4

Muestra un sumario de todos los subdirectorios bajo */home*, con una salida legible por las personas:

```
# du -csh /home/*
42k /home/bsmith
1.5M /home/httpd
9.5M /home/jdean
42k /home/jdoe
12k /home/lost+found
1.0k /home/samba
11M total
```

Este resultado muestra que se han utilizado 11 MB del espacio total del disco.

Ejemplo 5

Muestra el mismo sumario, pero ordenando los resultados de mayor a menor utilización:

```
# du -cs /home/* | sort -nr
11386 total
9772 jdean
1517 httpd
42 jdoe
42 bsmith
12 lost+found
1 samba
```

Este resultado muestra que el usuario *jdean* está consumiendo la mayor cantidad de espacio. Ten en cuenta que el formato legible por las personas no ordenaría de esta forma ya que *sort* no interpreta dicho formato.

Comprobando la Integridad del Filesystem



Independientemente de lo estables que sean, los ordenadores terminan fallando, incluso por algo tan simple como un cable de alimentación desconectado por accidente. Desafortunadamente una interrupción de este tipo puede provocar daños en un filesystem. Si se abortase una operación de escritura en disco antes de completarse, los datos implicados se perderían y las partes del disco que se reservaron para ellos quedarían marcadas como en uso. Además, las escrituras en el filesystem no suelen ser directas, sino que suelen pasar previamente por una cache de memoria, un corte de corriente o una caída del sistema impediría al kernel sincronizar ésta cache con el disco. Cualquiera de estos casos provocaría que nos encontrásemos con inconsistencias en el filesystem que deberán ser corregidas para asegurar un funcionamiento fiable del mismo.

Los filesystems se comprueban con *fsck*. Al igual que *mkfs*, *fsck* es una utilidad específica al tipo de filesystem instalado – esto incluye *fsck.ext2*, que es un link al programa *e2fsck* (mira en las páginas

man para más información).

Una parte de la información almacenada en el disco para describir un filesystem es la conocida como *superbloque* que se encuentra en el bloque 1 de la partición. Si éste área se corrompiese el filesystem quedaría inaccesible. Debido a la importancia del superbloque, se realizan copias del mismo en intervalos regulares del filesystem, por defecto cada 8192. La primera copia del superbloque se encuentra en el bloque 8193, la segunda en el bloque 16385, y así sucesivamente. Como verás, *fsck* puede utilizar la información en las copias del superbloque para restaurar el superbloque principal

Sintaxis

```
fsck [opciones] [-t tipo] [opciones-fs] filesystems
```

Descripción

Comprueba si los *filesystems* tienen errores y, opcionalmente, los corrige. Por defecto, *fsck* asume el tipo de filesystem *ext2* y funciona de modo interactivo interrumpiendo la ejecución para pedir permiso antes de aplicar las correcciones.

Durante la comprobación del sistema de *fsck* se hace lo siguiente:

1. Comprobar inodos, bloques y tamaños.
2. Comprobar la estructura de directorios.
3. Comprobar la conectividad de directorios.
4. Comprobar las referencias.
5. Comprobar el total de la información.



Tabla 2-3 Opciones frecuentemente utilizadas en *fsck*

Opción	Uso
-A	Ejecuta comprobaciones en todos los filesystems incluidos en <i>/etc/fstab</i> . Esta opción está pensada para utilizarse en tiempo de carga del sistema, antes de montar los filesystems.
-N	No se ejecuta, pero muestra lo que debería hacerse.
-t tipo	Especifica el tipo de filesystem a comprobar; por defecto se asume <i>ext2</i> . El valor de <i>tipo</i> determina que verificador específico para el filesystem es utilizado.
-b superbloque	Utiliza una copia del <i>superbloque</i> alternativa. En el modo interactivo, <i>e2fsck</i> utiliza automáticamente superbloques alternativos. Normalmente, para restaurar un superbloque defectuoso, utilizarás <i>-b 8193</i> en el modo no interactivo.
-c	Comprobar bloques defectuosos.
-f	Fuerza una comprobación, incluso si el filesystem parece limpio.
-p	Repara automáticamente el filesystem sin hacer preguntas.
-y	Responde automáticamente "yes" a todas las preguntas interactivas permitiendo la utilización no interactiva de <i>e2fsck</i> .

Ejemplo 1

Comprueba el filesystem del tipo *ext2* en */dev/hda5* que, en este momento, no está montado:

```
# fsck /dev/hda5
[/sbin/fsck.ext2 -- ] fsck.ext2 /dev/hda5
Parallelizing fsck version 1.14 (9-Jan-1999)
e2fsck 1.14, 9-Jan-1999 for EXT2 FS 0.5b, 95/08/09
/dev/hda5: clean, 1011/34136 files, 4360/136521 blocks
```

La partición estaba marcada como limpia, por tanto *fsck* no llegó a verificarla.

Ejemplo 2

Fuerza una comprobación:

```
# fsck -f /dev/hda5
Parallelizing fsck version 1.14 (9-Jan-1999)
e2fsck 1.14, 9-Jan-1999 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda5: 1011/34136 files (0.1% non-contiguous),
4360/136521 blocks
```

Ejemplo 3

Fuerza otra comprobación, en este caso con salida de mensajes explícita:

```
# fsck -fv /dev/hda5
Parallelizing fsck version 1.14 (9-Jan-1999)
e2fsck 1.14, 9-Jan-1999 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
 1011 inodes used (2%)
   1 non-contiguous inodes (0.1%)
   # of inodes with ind/dind/tind blocks: 0/0/0
4360 blocks used (3%)
   0 bad blocks
1000 regular files
   2 directorios
   0 character device files   0 block device files
   0 fifos
   0 links
   0 symbolic links (0 fast symbolic links)
   0 sockets
-----
 1002 files
```

Ejemplo 4

Permite que *fsck* realice automáticamente todas las reparaciones en un filesystem dañado especificando la opción *-y*:

```
[root@smp /mnt]# fsck -y /dev/hda5
Parallelizing fsck version 1.14 (9-Jan-1999)
e2fsck 1.14, 9-Jan-1999 for EXT2 FS 0.5b, 95/08/09
Couldn't find ext2 superblock, trying backup blocks...
/dev/hda5 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences: +1 +2 +3 +4
Fix? yes

Inode bitmap differences: +1 +2 +3 +4 +5 +6
Fix? yes

/dev/hda5: ***** FILE SYSTEM WAS MODIFIED *****
/dev/hda5: 1011/34136 files (0.1% non-contiguous),
4360/136521 blocks
```

Cuando Linux carga, el kernel realiza una comprobación de todos los filesystems incluidos en */etc/fstab* utilizando la opción *-A*. Se comprobará cualquier filesystem que no hubiese sido desmontado limpiamente (A menos que la entrada en */etc/fstab* incluya la opción *noauto*) Si dicha comprobación encontrase algún error significativo, el sistema se pondría en modo monousuario de tal forma que se pueda ejecutar *fsck* de forma manual.

Algunos de los errores que pueden causar esto son:

- Bloques solicitados por múltiples ficheros.
- Bloques solicitados fuera del sistema de ficheros.
- Detectados pocos enlaces.
- Bloques no detectados.
- Directorios que corresponden a inodos no localizados.
- Errores de formato.

En los casos dónde el directorio padre de un fichero no pueda ser determinado, el fichero será ubicado en */lost+found*. Los ficheros entonces se renombran con su número de inodo. Es útil examinar el contenido de este directorio después de haber perdido ficheros a consecuencia de un error del sistema.

La información (códigos) de finalización que nos da la utilidad *fsck* es útil para determinar el resultado de la operación. Cada código representa un tipo de condición de finalización. El código que nos retorna es la suma de las condiciones de salida. Los códigos de salida se muestran en la línea de comandos cuando el comando finaliza su operación. Estos códigos se muestran en la Tabla 2-4.

Tabla 2-4 Códigos de Finalización de fsck

<i>Código</i>	<i>Significado</i>
0	Sin error.
1	Errores del sistema de ficheros corregidos.
2	El sistema debería ser reiniciado.
4	Errores del sistema de ficheros sin corregir.
8	Error operacional.
16	Errores de sintaxis o uso.
128	Error en la librería compartida.

Desafortunadamente, a menos que tengas un conocimiento muy detallado del funcionamiento interno del filesystem, podrás hacer muy poco aparte de permitir que *fsck* realice todas las reparaciones. Por todo esto, lo más normal es utilizar la opción *-y* y confiar en la suerte.



En el Examen

Es importante que estés familiarizado con *du*, *df*, y *fsck*. Asegurarse de conocer las diferencias entre éstos comandos y cuando debe ser utilizado cada uno.

Creando un sistema de ficheros



Para crear un sistema de ficheros debe ser utilizada la herramienta correcta de las especificadas en la tabla 2-5. Un ejemplo de uso correcto de esas utilidades es crear una partición *ext2* utilizando *mkfs.ext2* tal como podemos ver:

```
# mke2fs /dev/hda3
```

Tabla 2-5 Utilidades para la creación de filesystems

<i>Comando</i>	<i>Tipo de filesystem creado</i>
<i>mkfs.ext2</i> o <i>mke2fs</i>	<i>ext2</i>
<i>mkfs.msdos</i> o <i>mkdosfs</i>	MD-DOS
<i>mkswap</i>	<i>swap</i>
<i>mkraid</i>	Raid
<i>mkfs.mimix</i>	<i>mimix</i>
<i>mkfs.bfs</i>	SCO BFS

Algunas utilidades de los filesystems



Comando debugfs – Depurador de filesystems (sistemas de ficheros) ext2.

Sintaxis:

```
debugfs [ -b tamaño_bloque ] [ -s superbloque ] [ -f fichero_comandos ] [ -R peticion ] [ -V ]  
[ [ -w ] [ -c ] [ -i ] [ dispositivo ] ]
```

Descripción

El programa debugfs es un depurador interactivo de filesystems. Puede utilizarse para examinar y cambiar el estado de un filesystem del tipo ext2. Dispositivo indica el fichero especial correspondiente al dispositivo que contiene el filesystem ext2 (p.ej. /dev/hdXX).



Comando dumpe2fs – volcado de la información de un filesystem

Sintaxis:

```
dumpe2fs [ -bfhixV ] [ -ob superbloque ] [ -oB tamaño_bloque ] dispositivo
```

Descripción:

dumpe2fs muestra la información de grupo del superbloque y los demás bloques del filesystem existente en dispositivo.



Comando tune2fs - ajusta los parámetros configurables en un filesystem ext2.

Sintaxis:

```
tune2fs [ -l ] [ -c contador-max-montajes ] [ -e comportamiento-errores ] [ -f ] [ -i intervalo-  
entre-comprobaciones ] [ -j ] [ -J opciones-de-diario ] [ -m porcentaje-bloques-reservados ] [ -o  
[^]opciones-de-montaje[,...] ] [ -r contador-bloques-reservados ] [ -s sparse-super-flag ] [ -u  
usuario ] [ -g grupo ] [ -C contador-montajes ] [ -L nombre-volumen ] [ -M directorio-ultimo-  
montaje ] [ -O [^]caracteristica[,...] ] [ -T hora-ultima-comprobacion ] [ -U UUID ]dispositivo
```

Descripción:

tune2fs ajusta los parámetros configurables en un filesystem ext2.

Tema 104.3

Montando y

desmontando

filesystems

Introducción

En este capítulo se verá como montar y desmontar sistemas de ficheros para poder trabajar con ellos de forma normal. Hoy en día existen herramientas que automatizan este proceso y esconden en cierta forma el trabajo del montado/desmontado. También se verá la estructura del fichero `/etc/fstab` en el que se guardan las particiones del disco duro, su formato, permisos, etc.

Los comandos que se verán en este tema son:

mount
umount

Y la estructura y funcionalidad de
`/etc/fstab`

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Controlando el montaje / desmontaje de los Filesystems

Como ya se vió anteriormente, la estructura de los sistemas de ficheros en GNU/Linux están generalmente divididos en particiones, unidas todas ellas en el punto de montaje raíz (/). Los filesystems de los dispositivos removibles, tales como CD-ROM, diskettes, discos ZIP, etc, se unen a la raíz del sistema de la misma manera, como directorios (o puntos de montaje). En principio estos directorios destinados a los dispositivos están vacíos, a la espera de su montaje, puede darse el caso de que el directorio destinado a este fin contenga subdirectorios o archivos, en cuyo caso quedarán ocultos hasta que el dispositivo se desmonte.

Administrando la tabla de filesystems



Para que las diferentes particiones estén disponibles desde un primer momento es necesario montarlas durante el inicio del sistema, los dispositivos removibles también se usan frecuentemente y es aconsejable tenerlos preparados para usar los comandos de montaje. Toda esta información se guarda en el fichero `/etc/fstab`. Los filesystems definidos en este fichero son revisados y montados durante el arranque del sistema. Sus entradas se consultan como fuente de información por defecto cuando los usuarios quieren montar dispositivos removibles.

En el siguiente ejemplo de `/etc/fstab` se puede ver que se trata de un fichero de texto con 6 campos en cada línea:

Device	Mount point	F. type	M. Options	DF	PN
/dev/sda1	/	ext2	defaults	1	1
/dev/sda5	/boot	ext2	defaults	1	2
/dev/sda9	/home	ext2	defaults	1	2
/dev/sda10	/tmp	ext2	defaults	1	2
/dev/sda11	swap	swap	defaults	0	0
/dev/fd0	/mnt/floppy	ext2	noauto,users	0	0
/dev/hdc	/mnt/cdrom	iso9660	noauto,ro,users	0	0

Device: Este campo especifica la partición del filesystem, por ejemplo `/dev/hda1`. Este campo no puede contener el dispositivo entero (`/dev/hda`)

Mount point: Aquí se introduce el directorio donde se quiere que el dispositivo sea montado. Por ejemplo si la partición `/dev/hda1` tiene el filesystem root, donde está el sistema se montará en `/`

Filesystem type: En este campo se indica que tipo de partición se trata, `ext2`, `reiserfs`, `swap`, `iso9660` (CD-ROMS) etc etc

Mount options: Se explican más adelante, se separan por comas.

Dump frequency: El programa `dump` (para hacer backups) consulta la entrada del `/etc/fstab` para ver cada cuanto tiempo debe hacer el backup. Normalmente tiene el valor 1 para filesystems basados en `ext2` y 0 para los demás.

Pass number para el fsck: Este campo es usado por la utilidad `fsck` cuando la opción `-A` se especifica, normalmente en tiempo de arranque. Tiene valores:

- 0 - No chequear el sistema
- 1 - Debe ponerse para el filesystem root (/) y ordena al fsck a chequearlo primero
- 2 - Hace el chequeo de la unidad, después del chequeo de los marcados con un 1

En el ejemplo se tiene un disco duro SCSI (dev/sda) La primera partición /dev/sda1 tiene el directorio root (/), la quinta contiene las imágenes del kernel para el arranque, la 9 para el directorio de los usuarios, la partición 10 es la temporal, la partición 11 para el sistema de swap. Luego se tiene la /dev/fd0 para diskettes y la /dev/hdc para el CD-ROM

Se pueden añadir/modificar estas entradas en cada caso y según las necesidades de cada uno.



Las opciones de este fichero son importantes de cara al examen.

Montando filesystems

Los filesystems son montados con el comando mount. Durante el arranque, los filesystems que no contienen un 0 en el pass number son chequeados y luego montados. Después del arranque se pueden añadir más sistemas de ficheros manualmente con el comando mount.

Sintaxis

```
mount [opciones] device
mount [opciones] directorio
mount [opciones] device directorio
```



Descripción: Se usa para montar (y así poder usar) filesystems dentro de la estructura del árbol del sistema. La primera y segunda entrada consultan al fichero /etc/fstab para montar los dispositivos y así tomar las opciones que se le especifiquen en el /etc/fstab. La tercera opción es independiente del fichero /etc/fstab y monta el filesystem (device) en el directorio (directorio)

El comando mount admite dos tipos de opciones, unos para el comando en si, y otros para especificar opciones del sistema de ficheros.

Tabla 3-1 Opciones del comando mount

<i>Opción</i>	<i>Función</i>
-a	Monta todos los filesystems especificados en el /etc/fstab menos los que tengan la opción noauto
-h	Ayuda del comando mount
-o	Especifica las opciones del mount en la linea de comandos
-r	Monta filesystems en modo de solo lectura
-t fstype	Especifica un tipo de filesystem
-v	Salida interactiva
-w	Monta filesystems de lectura/escritura

Opciones del mount

Estas opciones se especifican en el fichero /etc/fstab o bien en la línea de comandos con la opción -o. Estas opciones modifican el modo de montaje del comando mount, algunas de las opciones son:

Tabla 3-2 Opciones del mount sobre el sistema de ficheros

<i>Opción</i>	<i>Función</i>
async	Toda la E/S al sistema de ficheros debería hacerse asíncronamente.
auto	Puede montarse con la opción -a
defaults	Establece las opciones: rw, suid, dev, exec, auto, nouser y async. Es la opción por defecto en sistemas ext2
dev	Interpretar dispositivos especiales de caracteres o bloques en el sistema de ficheros
exec	Permitir la ejecución de binarios
noauto	Sólo puede montarse explícitamente (esto es, la opción -a no hará que el sistema de ficheros se monte)
noexec	No permitir la ejecución de ningún binario en el sistema de ficheros montado. Esta opción puede ser útil para un servidor que tiene sistemas de ficheros que contienen binarios para otras arquitecturas distintas de la suya.
nosuid	No permitir el efecto de los bits SUID ni SGID
nouser	Prohibir a un usuario ordinario (esto es, distinto de root) montar el sistema de ficheros. Esto es lo predeterminado
ro	Montar el sistema de ficheros en modo de sólo lectura.
rw	Montar el sistema de ficheros de lectura y escritura
suid	Permitir el efecto de los bits SUID y SGID
sync	Toda la E/S al sistema de ficheros debería hacerse síncronamente.
user	Permitir a un usuario ordinario montar el sistema de ficheros
users	Permite a cualquier usuario el montaje/desmontaje de el sistema de ficheros

Ejemplo 1: Para mostrar los filesystems actualmente montados en el sistema

```
$ mount
```

Ejemplo 2: Montar un CD-ROM situado en /dev/hdc en el directorio existente /cdrom de sólo lectura:

```
$ mount -rt iso9660 /dev/hdc /cdrom
```

Si se monta sin la opción -r, se mostrará un aviso de que el CD-ROM está protegido contra escritura y se montará como sólo lectura, sería el caso del comando:

```
$ mount -t iso9660 /dev/hdc /cdrom
```

Ejemplo 3: Montar un diskette (/dev/fd0) con sistema MS-DOS en el directorio /floppy

```
$ mount -t msdos /dev/fd0 /floppy
```

Ejemplo 4: Si da el caso de que la partición /home y /swap fueron desactivadas por algún motivo se podrían volver a montar con la opción -a

```
$ mount -av
```

De este modo montamos los filesystems que no lo estén y nos muestra la acción del comando conforme se va ejecutando (opción -v)

Desmontando filesystems



Los filesystems pueden ser desmontados usando el comando umount. Cuando un filesystem es desmontado, los contenidos del árbol principal se actualizan, no pudiéndose usar (el umount) si el sistema de ficheros que se quiere desmontar está en uso.

Si el sistema de ficheros está en uso el comando umount dará un error. Esto puede ocurrir por ejemplo cuando tenemos abierto un fichero de un CD-ROM o un proceso está haciendo uso del mismo. Otros errores pueden surgir si quitamos dispositivos removibles sin antes desmontarlos: pérdida de datos, corrupción de los mismos etc etc

Sintaxis:

```
umount [opciones] device
umount [opciones] directorios
```

Descripción: Desmonta un filesystem de un dispositivo o un directorio.

Tabla 3-3 Opciones del comando umount

Opción	Uso
-a	Desmonta todos los filesystems descritos en /etc/mstab. Este fichero está mantenido por los comando mount y umount en tiempo real, se usa normalmente cuando se apaga/reinicia el PC.
-t fstype	Desmonta sólo los filesystems del tipo especificado

Ejemplo 1: Desmontar el cdrom (/dev/hdc) montado en /cdrom

```
$ umount /cdrom
```

o bien

```
$ umount /dev/hdc
```

Ejemplo 2: Desmontar todos los sistemas de ficheros NFS:

```
$ umount -at nfs
```

Tema 104.4

Administrando

cuotas de disco

Introducción

En este capítulo se verá como administrar las cuotas en el disco para los usuarios. Entre los objetivos está el establecimiento, editado, comprobación y generación de informes

Los comandos que se verán en este tema son:

- quota
- edquota
- repquota
- quotaon
- quotaoff
- quotacheck

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Establecer y Consultar Cuotas de Disco



La administración del espacio en disco puede ser una tarea problemática. El espacio disponible es un recurso limitado que muy a menudo se consume a un ritmo alarmante provocando que un filesystem bien proporcionado se vuelva insuficiente al cabo del tiempo. En los sistemas multiusuario – no importa lo grande que sea el filesystem – los usuarios siempre terminarán llenándolo. Y a nadie le interesa que un filesystem se llene demasiado pronto. Una forma de prevenir que ésto ocurra es el establecimiento de cuotas de disco, de esta forma podrán establecerse límites a la cantidad de espacio en disco consumido por usuarios independientes o por grupos de éstos.

Típicamente un tamaño de cuota es mucho menor que el espacio libre del filesystem donde se configura, de esta forma se puede evitar que un usuario o un grupo consuma demasiado espacio. Pueden configurarse cuotas para cada filesystem incluido en `/etc/fstab`, aunque normalmente solo se aplican en aquellos filesystems donde los usuarios finales guardan sus archivos (p.e., `/home/nombreusuario`). No es necesario establecer una cuota en `/usr`, por ejemplo, ya que los usuarios finales no pueden guardar ficheros allí. Pueden configurarse cuotas para los usuarios individuales incluidos en `/etc/passwd` y para los grupos en `/etc/group`.

Cuotas de Limitación



Pueden establecerse hasta cinco tipos de cuotas de limitación por cada filesystem. Estas limitaciones se especifican en bloques de disco, normalmente de 1024 bytes cada uno:

Límite “hard” por usuario:

El límite hard es la máxima cantidad de espacio en disco que un usuario puede disponer en el sistema. Una vez que el usuario alcanza este límite ya no se le permitirá realizar nuevas escrituras en el disco.

Límite “soft” por usuario:

Cada usuario puede almacenar datos libremente en el filesystem hasta que alcance el límite soft. Éste límite actúa como una especie de zona de aviso, advirtiendo al usuario que debe ir limpiando sus directorios pero, a diferencia del límite hard se le permite seguir trabajando. Cuando el espacio consumido por un usuario supera el límite soft pero no el límite hard se envían mensajes de aviso al terminal del usuario advirtiéndole que está excediendo su cuota pero las operaciones de escritura terminarán correctamente.

Límite “hard” por grupo:

Se trata del límite final asignado a un grupo por el sistema de cuotas. Una vez que se alcanza este límite, ninguno de los miembros de dicho grupo podrán realizar escrituras en el disco, incluso aunque no hayan excedido sus límites individuales.

Límite “soft” por grupo:

Este límite se comporta de la misma forma que el de usuario pero se controla en base al espacio consumido por el grupo en lugar de por el usuario individual.

Periodo de gracia:

Una vez que se alcanza el límite soft comienza el periodo de gracia para el usuario o el grupo. Cuando expira éste periodo de gracia, el límite soft se convierte en límite hard hasta que se hayan eliminado los ficheros suficientes para que se resuelva el exceso de cuota. El periodo de gracia puede ajustarse para cualquier número de meses, semanas, días, horas, minutos o segundos. Un valor típico es el de siete días.

Todos estos límites pueden establecerse utilizando el comando `edquota`, que se detallará a continuación.

Cuando una escritura en disco excede un límite hard o un límite soft con el periodo de gracia expirado, solo se completará una parte de la operación de escritura dejando un fichero truncado y, probablemente, inservible.

Puede que el usuario no haya podido ver los mensajes de fin de cuota porque la aplicación haya ocultado la shell y los mensajes de dicha aplicación lo lleven a confusión al indicarle que el disco está lleno o protegido de escritura.

Comandos de cuotas

Linux proporciona una serie de comandos para gestión y consulta de las cuotas en los filesystems. Parte de la configuración requerida para establecer las cuotas inicialmente ha de hacerse a mano y sin ninguno de los comandos específicos de cuotas. Este proceso se explicará en la siguiente sección: [Habilitando las Cuotas](#).

quota



Muestra las limitaciones de cuota en usuario o en grupo. La opción `-u` está activada por defecto. Solo el superusuario puede utilizar la opción `-u` y usuario para ver las limitaciones de otros usuarios. El resto de los usuarios usuarios podrán usar la opción `-g` y grupo para ver solamente las limitaciones de los grupos de los que sean miembros, siempre y cuando tengan permiso de lectura en los ficheros `quota.group`.

Sintaxis:

`quota [-u] [opciones] usuario`

`quota -g [opciones] grupo`

Tabla 4-1 Opciones de uso frecuente del comando `quota`

<i>Opción</i>	<i>Función</i>
<code>-q</code>	Establece el modo “silencioso” que solo muestra las situaciones de exceso de cuota.
<code>-v</code>	Establece el modo explícito que muestra las cuotas incluso cuando no se ha consumido ningún espacio en disco o no se hayan definido.

Ejemplo 1

Siendo root, examina todas las cuotas del usuario fulanito :

```
# quota -uv fulanito
```

```
Disk quotas for user fulanito (uid 500):
Filesystem blks quota limit grace files quota limit grace
/dev/sda9  9456 10000 10200    32  0  0
/dev/hda1   23   0  0      17  0  0
```

Este ejemplo muestra que fulanito está cerca de consumir su límite soft de 10,000 bloques, con un límite hard de 10,200 bloques en /dev/sda9, y que no tiene cuota establecida en /dev/hda1. La línea correspondiente /dev/hda1 se muestra debido a la utilización de la opción -v. No se muestran los valores correspondientes al periodo de gracia porque aun no se ha excedido el límite soft.

Ejemplo 2

Siendo el usuario fulanito, examina las cuotas para el grupo finanzas, del que dicho usuario es miembro:

```
$ quota -gv finanzas
```

```
Disk quotas for group finanzas (gid 501):
Filesystem blks quota limit grace files quota limit grace
/dev/sda9  1000* 990  1000 6days  34 3980 4000
/dev/hda1    0   0  0      0  0  0
```

En este caso, el grupo finanzas ha excedido el escaso límite soft de 990 bloques y ha alcanzado su límite hard de 1000 bloques. (Probablemente no se pudo completar la operación de escritura que escribió el bloque número 1000.) El periodo de gracia original en este ejemplo se estableció en siete días de los cuales quedan seis días restantes, lo que quiere decir que ha pasado un día desde que se excedió el límite soft.

quotaon

Activa las cuotas configuradas previamente en uno o más filesystems.



Sintaxis:

```
quotaon [opciones] [filesystems]
quotaon [opciones] -a
```

Tabla 4-2 Opciones de uso frecuente del comando quotaon

<i>Opción</i>	<i>Función</i>
-a	Activa las cuotas en todos los filesystems incluidos en /etc/fstab y que estén marcados como de lectura-escritura y con cuotas. Normalmente se utiliza en el proceso de arranque para activar las cuotas.
-g	Activa las cuotas de grupo. Esta opción no es necesaria si se utiliza -a ya que ésta otra activa tanto las cuotas de usuario como las de grupo.
-u	Activa las cuotas de usuario; esta es la opción por defecto.
-v	Establece el modo explícito de tal forma que se muestra un mensaje por cada filesystem en el que se han activado las cuotas.

Ejemplo 1

Activar todas las cuotas definidas en /etc/fstab :

```
# quotaon -av
```

Ejemplo 2

Activar las cuotas de usuario sólo en el filesystem /home:

```
# quotaon -uv /home
```

quotaoff



Desactiva las cuotas de disco en uno o más filesystems.

Sintaxis:

```
quotaoff [opciones] [filesystems]
quotaoff [opciones] -a
```

Tabla 4-3 Opciones de uso frecuente del comando quotaoff

<i>Opción</i>	<i>Función</i>
-a	Desactiva las cuotas en todos los filesystems en /etc/fstab.
-g	Desactiva las cuotas de grupo. Esta opción no es necesaria si se utiliza la -a ya que ésta última incluye tanto las cuotas de usuario como las de grupo.
-u	Desactiva las cuotas de usuario; esta es la opción por defecto.
-v	Establece el modo explícito que mostrará un mensaje para cada filesystem en el que se desactiven las cuotas.

Ejemplo

Desactivar todas las cuotas mostrando todos los mensajes:

```
# quotaoff -av
```

quotacheck



Examina los filesystems y compila las bases de datos de cuotas. Esta comando no está incluido específicamente en los objetivos del examen LPI 101, pero es un componente importante en el sistema de cuotas de Linux. El comando quotacheck -a debería ser ejecutado de forma regular (posiblemente semanalmente) por medio de cron.

Sintaxis:

```
quotacheck [opciones] filesystems
```

```
quotacheck [opciones] -a
```

Tabla 4-4 Opciones de uso frecuente del comando quotacheck

<i>Opción</i>	<i>Función</i>
-a	Comprueba todas las cuotas de los filesystems incluidos en /etc/fstab. Se comprobarán tanto las cuotas de usuario como las de grupo según se indique en las opciones usrquota y grpquota.
-g grupo	Compila solamente la información de grupo.
-u usuario	Compila solamente la información de usuario; ésta es la opción por defecto. No obstante, si se especificase la opción -g, entonces habría que incluir esta si se desea que se procesen también las cuotas de usuario.
-v	Establece el modo explícito que mostrará toda la información de lo que el programa está haciendo. Esta opción indica que el programa está activo mediante un símbolo giratorio en el terminal. Ésto queda muy bonito pero podría ser un problema si estamos conectados mediante un módem lento.

Ejemplo 1

Inicializa todos los ficheros de cuotas:

```
# quotaoff -a
```

```
# quotacheck -aguv
```

```
# quotaon -a
```

Para actualizar los ficheros de bases de datos de cuotas debemos desactivar primero las mismas.

Ejemplo 2

Con las cuotas activas, actualizar las cuotas de usuario en memoria para el filesystem /home :

```
# quotacheck -v /home
```

edquota



Modifica las cuotas de usuario o grupo. Este es un comando interactivo que utiliza un editor de texto para configurar los parámetros de cuotas para usuarios o grupos. Por defecto se utiliza el editor vi a menos que las variables de entorno EDITOR o VISUAL apunten a otro editor como Emacs. Cuando se envía el comando, se ejecuta un editor con un fichero temporal que contiene las opciones de cuotas. Al guardar el fichero temporal, el editor finaliza y los cambios se guardan en las bases de datos de cuotas.

En la primera forma del comando, se modifica una lista de usuarios o grupos separados por espacios en blanco especificados en el apartado nombres. Si se incluye la opción -p y proto-usuario, las cuotas de ese usuario o grupo se utilizarán para los nombres y no se lanzará ningún editor.

En la segunda forma del comando, con la opción -t, se editarán interactivamente los límites soft para cada filesystem.

Sintaxis:

```
edquota [-p proto-usuario] [opciones] nombres
edquota [opciones] -t
```

Tabla 4-5 Opciones de uso frecuente del comando edquota

<i>Opción</i>	<i>Función</i>
-g	Modifica las cuotas de grupo. Si se especifica -g, se asumirá que todos los nombres son grupos y no usuarios, incluso si se especificase también -u.
-p proto-usuario	Duplica las cuotas del usuario prototipo para cada grupo o usuario especificado. Este es el mecanismo mas habitualmente utilizado para inicializar al mismo tiempo las cuotas de varios usuarios o grupos.
-t	Modifica los límites soft. Este comando interpreta unidades de sec (segundos), min(minutos), hour (horas), day (días), week (semanas), y month (meses).
-u	Modifica las cuotas de usuario. Es la opción por defecto, pero se ignorará si se especificase la opción -g

Los siguientes ejemplos muestran la salida de los comandos utilizando el editor vi.

Ejemplo 1

Modifica las cuotas de usuario para fulanito :

```
# edquota -u fulanito
```

Quotas for user fulanito:

```
/dev/sda9: blocks in use: 87, limits (soft = 99900,
hard = 100000)
inodes in use: 84, limits (soft = 0, hard = 0)
/dev/hda1: blocks in use: 0, limits (soft = 0, hard = 0)
inodes in use: 0, limits (soft = 0, hard = 0)
"/tmp/EdP.auHTZJ0" 5 lines, 241 characters
```

En este ejemplo, a fulanito se le ha definido en /dev/sda9 un límite soft de 99.900 bloques, un límite hard de 100.000 bloques, y ningún límite de ficheros. No tiene ninguna limitación en /dev/hda1.

Ejemplo 2

Modificar los límites soft para usuarios en todos los filesystems:

```
# edquota -tu
```

Time units may be: days, hours, minutes, or seconds

Grace period before enforcing soft limits for users:

/dev/sda9: block grace period: 7 days,

file grace period: 3 days

/dev/hda1: block grace period: 7 days,

file grace period: 3 days

"/tmp/EdP.aiTShJB" 5 lines, 249 characters

En este caso se han establecido unos periodos de gracia de siete días para bloques (espacio en disco) y de tres días para ficheros (inodos).

repquota

Se utiliza para obtener un informe de la situación de las cuotas.



En la primera forma del comando, repquota mostrará un informe por usuario o por grupo de las cuotas de los filesystems indicados.

En la segunda forma, la opción -a hará que se muestre un sumario de todos los filesystems con cuotas. Este comando solo podrá ejecutarlo el usuario root, a menos que los ficheros de bases de datos de las cuotas tengan permisos de lectura universales. Para cada usuario se imprime el número de ficheros y el espacio en disco utilizados en ese momento junto con las cuotas creadas con edquota.

Sintaxis:

```
repquota [opciones] filesystems
```

```
repquota -a [opciones]
```

Tabla 4-6 Opciones de uso frecuente del comando repquota

<i>Opción</i>	<i>Función</i>
-a	Hace informes de todas las cuotas de todos los filesystems de lectura-escritura incluidos en /etc/fstab. Se mostrarán tanto las cuotas de usuario como las de grupo según se indique en las opciones usrquota y grpquota.
-g	Muestra las cuotas de grupos.
-u	Muestra las cuotas de usuarios; es la opción por defecto.
-v	Activa el modo explícito, el cual añade una cabecera descriptiva a la salida del comando.

Ejemplo

Informe de las cuotas de usuario para el filesystem /home:

```
# repquota -v /home
```

Habilitando las cuotas



Para poder utilizar las cuotas, en primer lugar hay que habilitarlas (El kernel debe estar compilado con soporte para cuotas. En el caso improbable de que no fuese así, habría que recompilarlo con esta opción.). No es un proceso dificultoso pero, desgraciadamente, tampoco es simple. Para aclarar un poco este proceso, en esta sección se incluirá un breve tutorial explicando como habilitar las cuotas de usuario y de grupo en el filesystem /dev/sda9 montado bajo /home. Hay que tener en cuenta que pueden habilitarse cuotas solo de usuario, de grupo o ambas según sea necesario.

1. Definir las opciones en /etc/fstab. En la línea que contiene el filesystem /home, añadir las opciones usrquota y grpquota a la opción default, como en este ejemplo:

```
/dev/sda9 /home ext2 defaults,usrquota,grpquota 1 2
```

Estas opciones indican a las utilidades de configuración de cuotas que particiones deben ser tratadas por dichas utilidades cuando éstas busquen en /etc/fstab.

2. Crear los ficheros quota.user y quota.group en la raíz del filesystem /home y dar permisos sólo para root:

```
# touch /home/quota.user /home/quota.group  
# chmod 600 /home/quota.user /home/quota.group
```

Estos dos ficheros son las bases de datos de las cuotas de usuarios y grupos. Cada filesystem con cuotas utilizará sus propias bases de datos. Al habilitar las cuotas, estos ficheros contendrán datos binarios (no son ficheros de texto). Si se quisiera que los usuarios puedan examinar las cuotas de los grupos a los que pertenecen, el fichero quota.group necesitará un permiso de 644 en lugar de 600.

3. Ejecutar quotacheck para inicializar las bases de datos:

```
# quotacheck -avug  
Scanning /dev/sda9 [/home] done  
Checked 236 directories and 695 files  
Using quotafile /home/quota.user  
Using quotafile /home/quota.group
```

4. Ahora, verificar que los ficheros de bases de datos se han inicializado realmente, para ello nos aseguramos que su tamaño no sea cero (en este ejemplo cada uno ocupa 16.192):

```
# ls -al /home/quota.*  
-rw----- 1 root root 16192 Dec 27 19:53 /home/quota.group  
-rw----- 1 root root 16192 Dec 27 19:53 /home/quota.user
```

5. Ejecutar quotaon para activar el sistema de cuotas:

```
# quotaon -a
```

6. Verificar que el script de inicialización del sistema (/etc/rc.d/rc.sysinit o similar) activará las

cuotas cada vez que el sistema cargue.

Puede ser algo similar al ejemplo siguiente aunque puede variar de sistema a sistema:

```
if [ -x /sbin/quotacheck ]
then
  echo "Checking quotas."
  /sbin/quotacheck -avug
  echo " Done."
fi
if [ -x /sbin/quotaon ]
then
  echo "Turning on quotas."
  /sbin/quotaon -avug
fi
```

7. Añadir un script al directorio crontab del sistema (por ejemplo /etc/crontab.weekly) para que se ejecute quotacheck rutinariamente. El script del ejemplo puede ser válido:

```
#!/bin/bash
/sbin/quotacheck -avug
```

Alternativamente, podría ponerse /sbin/quotacheck en el fichero crontab de root (utilizando el comando crontab -e) para su ejecución semanal, como en este ejemplo:

```
# ejecutar quotacheck semanalmente
0 3 * * 0 /sbin/quotacheck -avug
```

En este momento, el filesystem /home está listo para aceptar cuotas de usuario o de grupo, controlarlas y emitir informes sobre ellas.

Tema 104.5

Usando permisos para controlar el acceso a los ficheros

Introducción

En este capítulo se verá como controlar los accesos a los ficheros y directorios por medio de los permisos. También se hablará de bits especiales como el suid, sgid y sticky bit y usar permisos para grupos.

Los comandos que se verán en este tema son:

chmod
umask

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Permisos de Archivos y Directorios

Los permisos determinan quién puede acceder a los archivos y directorios dependiendo del tipo de acceso que tengan. Los primeros 10 caracteres de un listado `ls -l` de cualquier entidad se parecen a lo siguiente:

`-rwxrwxrwx`



El primer carácter se identifica con el tipo de entidad: `-` para un archivo estándar, `d` para un directorio, `b` para un grupo de recursos (tales como una unidad de cinta), `c` para un carácter del recurso, `l` para un link, o `p` para una tubería (pipe). El resto de los nueve caracteres se dividen en 3 grupos, como se indica en la Figura 8.1.



Cuando un usuario intenta acceder a un archivo, el primer control confirma si el es el propietario del archivo. Si lo es, se le aplica el primer tipo de permisos. Si no lo es, el segundo control confirma si es un miembro del grupo propietario del archivo. Si es un miembro del grupo, se le aplica el tipo intermedio de permisos. Si no es propietario del archivo, y no es miembro del grupo propietario, se le aplica el tercer tipo de permisos.

Figura 8.1 Permisos



Permisos estándar

Los permisos que se pueden aplicar a una entidad -propietario, grupo u otro- son:



- `r`- Permite la lectura de un archivo. Éste es el único permiso necesario para copiar un archivo. Cuando se aplica a un directorio, se pueden leer (ver) sus archivos.
- `w`- Permite escribir en un archivo. Con él se pueden cambiar, modificar o sobrescribir los contenidos del archivo. Cuando se aplica en un directorio, este permite borrar y mover archivos (incluso si no se tiene el permiso de escritura específico sobre el archivo individual).
- `x`- Permiso de ejecución: permite ejecutar el archivo si contiene los scripts necesarios o puede ser ejecutado por el sistema. Aplicado a un directorio, este permite el acceso al mismo. Cuando se aplica a un conjunto con permisos de lectura dentro de un escritorio, este permite buscar dentro de dicho directorio.
- `-` (guión)- Indica la ausencia de permiso. Por ejemplo, `r-x` indica que ese usuario puede leer y ejecutar, pero no escribir.

Por tanto, los 10 campos de permisos se resumen en:

- Tipo de entidad (archivo, directorio, otro)
- El propietario puede leer
- El propietario puede escribir
- El propietario puede ejecutar
- El grupo puede leer
- El grupo puede escribir
- El grupo puede ejecutar

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

- Usuario (no pertenece al grupo y al propietario) puede leer
- Usuario puede escribir
- Usuario puede ejecutar

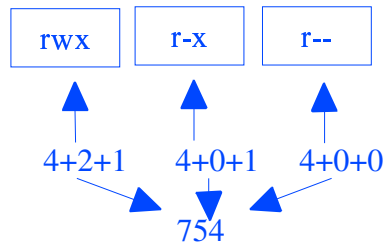
Estos permisos tienen valores numéricos como se muestran en la Tabla 5-1.

Tabla 5-1 Simbología de permisos y sus valores

<i>Permiso</i>	<i>Valor numérico</i>
r	4
w	2
x	1
-	0

Los valores numéricos hacen posible añadir permisos a la vez y expresarlo de un modo sencillo. Por ejemplo, si un archivo permite al usuario rwx, el valor numérico sería $4(r)+2(w)+1(x)=7$. El formato del conjunto de los permisos de un archivo se muestra en la Figura 8.2.

Figura 8.2 Valores numéricos para los permisos de archivo.



La tabla 5-2 muestra la conversión numérica de un conjunto de distintos permisos:

Tabla 5-2 Conversión numérica

<i>Valor numérico</i>	<i>Permisos</i>
1	-----X
2	-----W-
3	-----WX
4	-----r--
5	-----r-X
6	-----rW-
7	-----rWX
10	----X---
11	----X--X
22	----W--W-
33	----WX-WX
55	---r-Xr-X

Tema 104 Dispositivos, sistemas de ficheros y sus estándares

<i>Valor numérico</i>	<i>Permisos</i>
77	---rwxrwx
100	--x-----
101	--x-----x
111	--x--x--x
222	-w--w--w-
311	-wx--x--x
322	-wx-w--w-
400	r-----
444	r--r--r--
511	r-x---x--x
544	r-xr--r--
644	rw-r--r--
666	rw-rw-rw
755	rxrx-r-x
777	rxrxrwx



Los permisos por defecto para todos los nuevos archivos creados son 666 (rw-rw-rw-) y para los directorios son 777(rwxrwxrwx). Este número puede ser modificado mediante la variable umask. La variable umask indica la cantidad sustraída al permiso por defecto hasta llegar a los permisos que se le aplicarán al usuario.

Para ver el valor de umask, lo tecleamos en la línea de comandos:

```
$ umask
022
```

Con un umask de 022, los permisos asignados a los nuevos archivos serán 644 (rw-r--r-) y a los directorios 755 (rwxr-xr-x), como se muestra en la Figura 5.3:

Tabla 5-3 Cálculo de los valores de las nuevas entidades después de sustraer el valor de umask.

<i>Archivos</i>	<i>Directorios</i>
666 -rw-rw-rw- - 022 ----w--w-	777 dwxrwxrwx - 022 ----w--w-
644 -rw-r--r--	755 dwxr-xr-x

Se pueden cambiar los valores de umask especificando un valor diferente en la línea de comandos (umask 15, por ejemplo), y este valor es el usado para la sesión. La variable se define en la información de inicio de sesión y se recupera (toma el valor inicial) al comienzo de la misma.

Cambiando Valores



Para cambiar los permisos de un archivo o directorio, se puede usar la utilidad `chmod`. Los argumentos pueden ser números o letras. Por ejemplo, para modificar los permisos de un archivo que permita a todos leer y escribir en él, se deberá entrar lo siguiente:

```
$ ls -l turbo
-rw-r--r-- 1 root root 14 Sep 6 22:42 turbo
$ chmod 666 turbo
$ ls -l turbo
-rw-rw-rw- 1 root root 14 Sep 6 22:42 turbo
```

En formato simbólico, `u` significa usuario, `g` grupo y `o` es otro. Se puede elegir y añadir según los permisos existentes:

```
$ ls -l turbo
-rw-r--r-- 1 root root 14 Sep 6 22:42 turbo
$ chmod go+w turbo
$ ls -l turbo
-rw-rw-rw- 1 root root 14 Sep 6 22:42 turbo
```

o especificar los permisos directamente:

```
$ ls -l turbo
-rw-r--r-- 1 root root 14 Sep 6 22:42 turbo
$ chmod ugo=rw turbo
$ ls -l turbo
-rw-rw-rw- 1 root root 14 Sep 6 22:42 turbo
```

Se puede utilizar el signo `+` para añadir a los permisos existentes y el `-` para borrarlos. El signo `=` ignora la existencia de permisos y fija el valor indicado. La opción `-c` indica a `chmod` que devuelva los nombres de los archivos que han cambiado, y la `-f` elimina la visualización por pantalla de los mensajes de error.

Permisos especiales



Pueden ser utilizados 3 tipos de permisos en determinadas circunstancias. Aparte de los siempre aplicables de lectura, escritura y ejecución, algunas veces es necesario algo más para un archivo o directorio. Estos permisos especiales son los 3 siguientes:

- Asignar ID de usuario (set user ID) (SUID)
- Asignar ID de grupo (set group ID) (SGID)
- Sticky bit

SUID



La asignación de ID de usuario se aplica cuando se desea que un determinado usuario ejecute un programa que de otro modo no podría.

Por ejemplo, sólo el usuario root sería capaz de ejecutar la función `funcion xyz` (comenzar backups, restaurar el sistema, entrar en otros recursos, etc) a causa de las ramificaciones de seguridad, pero se necesita que los usuarios ejecuten un shell script para realizar esta acción, porque no se dispone del tiempo necesario para hacerlo personalmente.

Se puede crear este shell script como root y asignar el permiso SUID de modo que el usuario que ejecute el script sea root sólo dentro de ese script. Antes y después del manuscrito, es únicamente un usuario, pero durante la ejecución del script es como si fuera root.

El permiso numérico de SUID, 4000, es sumado al valor de otros permisos. Una vez aplicado este, cambia la x en el campo del ejecutable para el propietario de los permisos a una s:

```
$ ls -l turbo2
```

```
$ chmod 4777 turbo2
```

```
$ ls -l turbo2
```

Recordar: El objetivo de la utilización de este permiso es que el proceso sea ejecutado por la persona que lo creó (root en este caso) y no por la persona que lo ejecuta. Sintaxis:

```
chmod u+s turbo2
```

SGID



Similar en la naturaleza a SUID, el permiso de la identificación de grupo del sistema se aplica cuando es necesario que la persona que ejecuta el archivo sea un miembro del grupo que posee el archivo (y no el propietario). Esto cambia el x en el permiso del grupo a un s, y el valor numérico es 2000:

```
$ ls -l turbo2
```

```
$ chmod 2777 turbo2
```

```
$ ls -l turbo2
```

La sintaxis del comando es:

```
chmod g+s turbo2
```

Sticky Bit



Este permiso no trabaja como los otros permisos especiales. Con un valor numérico de 1000, sus operaciones difieren cuando están aplicadas a un directorio o a un archivo. Cuando está aplicado a un directorio, evita que los usuarios supriman archivos de las carpetas que les conceden el permiso de escritura, a menos que sean el propietario del archivo. Por defecto, cualquier usuario que tenga permiso de escritura en un directorio puede suprimir archivos dentro de ese directorio, incluso si no tiene el permiso de escritura de ese archivo.

Cuando se aplica sobre un archivo, el archivo se convierte en “sticky” (bloqueado). La primera vez que se accede o se ejecuta el archivo y se carga en memoria, permanece cargado en memoria física (RAM) o espacio swap de modo que pueda funcionar más rápidamente que si se lee desde el disco. Si el archivo no es ejecutable, el último bit de permiso (para otra categoría) se convierte en T. Si el archivo es un fichero ejecutable, o el permiso se aplica a un directorio, el bit pasado se convierte en una t. Cuando se aplica el permiso chmod y las letras, aparece t de todos modos (sea archivo o directorio).

Tema 104.6

Administrando el propietario de los ficheros

Introducción

En este capítulo se verá como cambiar el propietario de los ficheros y directorios.

Los comandos que se verán en este tema son:

chown
chgrp

Este tema tiene un peso (importancia) de 1 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Permisos de Usuarios y Grupos

Los ficheros y directorios usan 3 tipos de permisos, todos ellos se vieron en temas anteriores (propietario, grupo y otros). Esto permite restringir el acceso a los ficheros dependiendo del usuario y grupo que los use. Los sistemas GNU/Linux disponen de varias herramientas que permiten cambiar los permisos de acceso al fichero.

chown

El comando chown se usa para cambiar el propietario de los archivos o directorios, se usa con la siguiente sintaxis:

chown [opciones] usuario[:grupo] fichero



chown cambia el propietario de cada fichero, usuario o grupo, si sólo se da un nombre de usuario (o UID numérico), entonces ese usuario se convierte en el propietario de cada fichero dado, y el grupo al que pertenece el fichero no cambia. Si al nombre de usuario sigue un signo de dos puntos y un nombre de grupo (o GID numérico), sin espacios entre ellos, entonces también se cambia el grupo al que pertenece cada fichero.

Algunas de las opciones del comando chown se ven en la tabla 6-1

Tabla 6-1 Opciones del comando chown

Opción	Uso
-c	Muestra un mensaje donde menciona solamente aquellos ficheros cuyo propietario cambia realmente.
-f	No muestra mensajes de error sobre ficheros cuyo propietario no puede cambiarse.
-R	Cambia recursivamente el propietario de directorios y sus contenidos.
-v	Describe la acción efectuada (o no) para cada fichero de forma interactiva.

A continuación algunos ejemplos del uso del comando:

Para cambiar el propietario de un archivo, la secuencia es:

```
$ ls -l turbo
-rw-rw-rw- 1 root root 14 Sep 6 22:42 turbo
$ chown edulaney turbo
$ ls -l turbo
-rw-rw-rw- 1 edulaney root 14 Sep 6 22:42 turbo
```

Ejemplo 1: Asignar al usuario Angie el directorio stuff recursivamente:

```
$chown -R Angie stuff
```

Ejemplo 2: Asignar el usuario Angie todos los ficheros que comienzan por t_ y mostrar la salida interactiva, (todos los ficheros que empiezan por t_ ya pertenecen actualmente al usuario Angie)

```
$chown -v Angie t_*
```

el propietario de «t_manhunt_pc_gp2.wmv» permanece como Angie

el propietario de «t_manhunt_pc_gp.wmv» permanece como Angie



Es necesario recordar el hecho de que cambiar el propietario de un archivo, no cambia los permisos de los otros valores, sino que sólo cambia los permisos que no cambian para el propietario.

chgrp



El comando chgrp se usa para cambiar el grupo propietario de un fichero o directorio. El comando busca en el fichero /etc/group para confirmar la existencia del grupo especificado antes de cambiar los permisos. Las opciones más comunes se pueden ver en la tabla 6-2. El comando se usa con la siguiente sintaxis:

```
chgrp [opciones] grupo fichero
```

Tabla 6-2 Opciones del comando chgrp

Opción	Uso
-c	Muestra un mensaje donde menciona solamente aquellos ficheros cuyo propietario cambia realmente.
-h	Actúa sobre enlaces simbólicos propiamente en vez de sobre lo que apunten. Disponible solamente si el sistema proporciona la primitiva lchown.
-f	No muestra mensajes de error sobre ficheros cuyo grupo no pueda cambiarse.
-R	Cambia recursivamente el grupo al que pertenecen directorios y sus contenidos. (Y continúa incluso si se encuentran errores.)
-v	Describe la acción efectuada (o no) para cada fichero de forma interactiva.

El usuario root puede efectuar los cambios al grupo que desee; para que un usuario pueda hacerlo, debe ser el propietario del archivo y pertenecer al grupo donde está efectuando el cambio.

Si el usuario root cambia el propietario y el grupo al mismo tiempo, se puede utilizar chown y separar los dos valores por : (dos puntos):

```
$ ls -l turbo
```

```
-rw-rw-rw- 1 edulaney root 14 Sep 6 22:42 turbo
```

```
$ chown kristen:business turbo
```

```
$ ls -l turbo
```

```
-rw-rw-rw- 1 kristen business 14 Sep 6 22:42 turbo
```

El comando falla si el propietario o el grupo no existen. Se puede utilizar chown para cambiar el grupo usando sólo la segunda parte de los argumentos:

```
$ ls -l turbo
```

```
-rw-rw-rw- 1 kristen business 14 Sep 6 22:42 turbo
```

```
$ chown :users turbo
```

```
$ ls -l turbo
```

```
-rw-rw-rw- 1 kristen users 14 Sep 6 22:42 turbo
```

Tema 104.7

Crear y cambiar enlaces a ficheros

Introducción

En este capítulo se verá como crear y administrar enlaces fuertes y simbólicos a ficheros. Se verá como crearlos, identificarlos, copiar ficheros a través de enlaces y usar enlaces a ficheros para tareas administrativas

Los comandos que se verán en este tema son:

ln

Este tema tiene un peso (importancia) de 1 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Creando links a ficheros



A veces es útil o necesario tener un mismo fichero en varias localizaciones del sistema de ficheros, por ejemplo, para acceder con diferentes permisos sobre un fichero dependiendo de su localización. En Linux existe una solución para eso: los enlaces o links.

Estos son de dos tipos:

1. Los Duros o Hard Links, y
2. Los Suaves, Symbolic Links o Symlinks.

Los Hard Links

Estos enlaces comparten el inodo del fichero original. De hecho, un hard link es indistinguible del original y por eso los cambios en el link afectan al fichero original, excepto en el borrado. Borrar el link no elimina al original ni a la inversa. Este tipo de enlace conserva los permisos del original y marcas de tiempo. Por contra, no se pueden usar para hacer enlaces a directorios ni pueden extenderse a otros sistemas de ficheros. Su sintaxis es como sigue:

```
ln /ruta/completa/fichero nombre_enlace
```

Los Symlinks

Estos enlaces si que pueden extenderse a otros sistemas de ficheros. También pueden hacer referencia a directorios, de hecho, pueden referenciar hasta ficheros inexistentes. La lectura y escritura, así como la copia del enlace, afectan al fichero objetivo, mientras que el borrado afecta al propio enlace. Borrar el fichero objetivo tampoco elimina el enlace automáticamente. Su sintaxis es como sigue:

```
ln -s /ruta/completa/fichero nombre_enlace
```

Tema 104.8

Búsqueda de

ficheros

Introducción

En este capítulo se verá como encontrar ficheros en los filesystems de GNU/Linux, tener actualizada la base de datos, saber en que paquete está cada comando, etc.

Los comandos que se verán en este tema son:

- find
- locate
- slocate
- updatedb
- whereis
- which

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Buscando Archivos

Otra de las capacidades necesarias en un sistema Linux es la localización de archivos. Afortunadamente, Linux dispone de una gran variedad de métodos para buscar archivos. Los comandos **find**, **locate**, **which** y **where** son muy útiles para estas tareas.

find



La utilidad `find` se utiliza para buscar archivos. Esta herramienta comienza buscando por un directorio especificado y continúa por todos los subdirectorios que están bajo el mismo, en busca de nombres de archivos que coincidan con el patrón especificado. Cuando no se especifica ningún directorio, la utilidad `find`, comienza por el directorio actual (`pwd`) y continúa por todos los subdirectorios dentro de éste. La sintaxis correcta del comando es la siguiente:

```
find /directorio expresión
```

Las búsquedas por medio de la utilidad `find` pueden ser lentas y utilizar muchos recursos del sistema, dependiendo de la búsqueda requerida. La tabla 8-1 muestra algunas de las opciones que se pueden utilizar al usar el comando `find`. Algunos ejemplos serían:

```
# find /home/angie -name archivo
# find /home/angie -user root
```

Tabla 8-1 Opciones del comando `find`

Opción	Uso
<code>-atime</code>	Búsqueda basada en el número de días desde el último acceso.
<code>-ctime</code>	Búsqueda basada en el número de días desde el último cambio en la entrada del directorio.
<code>-group</code>	Búsqueda de archivos pertenecientes al grupo especificado.
<code>-newer</code>	Búsqueda de archivos más recientes que el archivo especificado.
<code>-name</code>	Búsqueda de archivos cuyo nombre coincida con la cadena especificada.
<code>-user</code>	Búsqueda de archivos pertenecientes al usuario especificado.

locate



El comando `locate` proporciona un método seguro para indexar y buscar rápidamente archivos en el sistema. Utiliza codificación incremental para comprimir su base de datos y hacer las búsquedas más veloces, pero también almacena los permisos y propietario del archivo, de modo que los usuarios que no puedan ver esos archivos, no podrán acceder a ellos. El comando `locate` utiliza por defecto la base de datos **slocate** para encontrar los archivos. Es posible especificar otra base de datos para ser usada. La sintaxis del comando es la siguiente:

```
locate -opciones argumentos
```



La base de datos se actualiza por medio del comando **updatedb**, sin argumentos. Sólo el usuario `root` está autorizado a utilizar este comando en el sistema.

El comando locate utiliza las opciones que se muestran en la tabla 8-2:

Tabla 8-2 Opciones del comando locate

<i>Opción</i>	<i>Uso</i>
-u	Comienza por el directorio raíz / cuando se crea la base de datos slocate.
-U /PATH	Comienza por el directorio especificado cuando se crea la base de datos slocate.
-e DIR	Excluye los directorios especificados cuando se crea la base de datos slocate.
-c	Analiza el archivo /etc/updatedb.conf cuando se crea la base de datos slocate.
-i	Busca sin tener en cuenta mayúsculas y minúsculas.
-o FILE	Especifica el archivo de salida a crear.
-d PATH	Especifica la ruta a las bases de datos a buscar.

which



El comando which toma uno o más argumentos. Para cada uno de sus argumentos, muestra por la salida estándar (stdout) la ruta completa a los ejecutables que se pueden ejecutar cuando dicho argumento se escribe en la línea de comandos. Ésto lo hace buscando ficheros ejecutables o scripts en los directorios especificados en la variable de entorno PATH, utilizando el mismo algoritmo que bash.

Esta utilidad nos permite ver la ruta completa a un comando antes de ejecutarlo. Puede ser útil para comprobar que estamos haciendo uso del comando que queremos ejecutar. La utilidad utiliza la siguiente sintaxis:

```
which -opciones nombre_de_programa
```

whereis



El comando whereis busca un archivo específico dentro de archivos fuente, binarios y páginas del manual. Los nombres especificados, antes de ser buscados, se separan de la definición de la ruta y de las extensiones del tipo .ext (por ejemplo .c). Finalmente, la utilidad whereis trata de encontrar los programas deseados dentro de las localizaciones de código. La tabla 8-3 resume las opciones utilizadas con whereis:

Tabla 8-3 Opciones del comando whereis

<i>Opción</i>	<i>Uso</i>
-b	Búsqueda de binarios.
-m	Búsqueda de entradas manuales.
-s	Búsqueda de fuentes.
-u	Búsqueda de entradas inusuales que no tienen una entrada por cada tipo.

El siguiente es un ejemplo del uso de whereis para localizar información sobre el comando ls:

```
# whereis ls
ls: /bin/ls /usr/man/man1/ls.1.gz
```

PREGUNTAS TEST

1. ¿Que opción, utilizada con *e2fsck*, especifica un superbloque alternativo cuando se usa para comprobar un filesystem?
A. -A
B. -b
C. -C
D. -l
2. El comando _____ muestra el espacio en disco utilizado por los filesystems montados.
3. ¿Cuántos pasos realiza la utilidad *fsck* para la comprobación de un filesystem?
A. Tres
B. Cuatro
C. Cinco
D. Seis
4. ¿Que comando se utiliza para consultar el espacio en disco utilizado dentro de un directorio?
A. df
B. du
C. mkfs
D. fsck
5. Antes de poder examinar un filesystem con *fsck*, primero ha de estar _____.
6. ¿Cuál de los siguientes permisos se representa por el valor numérico 44?
A. - - - - - r w -
B. - - - - r w - - - -
C. - - - - r - - r - -
D. - r - - r - - - -
7. ¿Cuales serán los permisos del archivo ejecutable “portable” cuando se utilice *chmod* con el valor numérico 1777?
A. - r w s r w x r w x
B. - r w x r w s r w x
C. - r w x r w x r w t
D. - r w x r w x t w T
8. ¿Que tipo de link puede apuntar a distintos sistemas de ficheros?
9. El comando *ln*, por defecto, creará _____.
10. ¿Cual de los resultados siguientes ocurre cuando se copia un soft link?
A. Se crea una nueva copia del soft link.
B. Se crea un hard link hacia el fichero original.
C. Se crea un hard link hacia el soft link.
D. Se crea una nueva copia del fichero original.

EJERCICIOS DE LABORATORIO

1. Crea un filesystem del tipo ext2 en la partición /dev/hda2.
2. Comprueba el filesystem utilizando *fsck*.
3. Consulta la información sobre el espacio libre y los inodos utilizando la utilidad *df*.

RESPUESTAS TEST

1. **B.** La opción `-b` se utiliza para especificar un superbloque alternativo con `fsck`. Para más información mira la sección “fsck”.
2. **df.** El comando `df` se usa para mostrar el espacio en disco utilizado por los filesystems. Para más información mira la sección “df”.
3. **C.** La utilidad `fsck` necesita cinco pasos para verificar el filesystem. Para más información mira la sección “fsck”.
4. **A.** El comando `du` se usa para ver la utilización del espacio en disco de los directorios. Para más información mira la sección “du”.
5. **desmontado.** Antes de poder inspeccionar un filesystem con `fsck`, éste ha de estar desmontado. Para más información mira la sección “fsck”.
6. La respuesta correcta a esta pregunta es la c. Cuando el valor numérico es inferior a cuatro dígitos, se asume que el resto son 0, por lo tanto 44 pasa a ser 0044 y los permisos son `----r--r--`. La respuesta a sería 6, la b 60 y la d sería equivalente a 440; por lo tanto, son incorrectas.
7. La respuesta correcta en este caso es la c. Debido a que el archivo es ejecutable, el último bit se convierte en t. La respuesta a es un valor igual a 4777, la b es 2777, y la d tiene como último bit T, lo que indica que no es ejecutable, es decir, también es incorrecta.
8. Los soft links pueden apuntar a diferentes sistemas de ficheros.
9. Hard link. Por defecto, el comando `ln` crea un hard link al fichero especificado. Para más información mira la sección “Hard links”.
- 10.D. Cuando se copia un soft link, se crea una nueva copia del fichero original en el destino especificado. Para más información mira la sección “Enlaces simbólicos”.

RESPUESTAS DE LABORATORIO

1. Para crear un filesystem `ext2` se utiliza el comando `mke2fs` junto con el nombre de dispositivo de la partición que lo contendrá.
`#mke2fs /dev/hda2`
2. Los filesystems de tipo `ext2` se comprueban con el comando `fsck.ext2`. La opción `-v` hace que la información sobre la ejecución sea más explícita y se muestran diferentes mensajes a medida que se va ejecutando la comprobación. Este comando requiere de un nombre de dispositivo, en este caso `/dev/hda2`.
`# fsck.ext2 -v /dev/hda2`
e2fsck 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts

```
Pass 5: Checking group summary information
9692 inodes used (0%)
157 non-contiguous inodes (1.6%)
# of inodes with ind/dind/tind blocks: 1194/205/0
1754206 blocks used (70%)
0 bad blocks
8884 regular files
776 directories
0 character device files
0 block device files
1 fifo
0 links
22 symbolic links (22 fast symbolic links)
0 sockets
-----
9683 files
```

3.

```
# df
Filesystem 1k-blocks Used Available Use%
Mounted on
/dev/hda8 4096380 1469176 2627204 36% /
/dev/hda5 15522 3710 11011 25% /boot
/dev/hda2 9740592 0 2384384 0% /fun
```

```
# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/hda8 3.9G 1.4G 2.5G 36% /
/dev/hda5 15M 3.6M 11M 25% /boot
/dev/hda2 9.3G 0G 9.3G 0% /fun
```

```
# df -i
Filesystem Inodes Iused Ifree Iuse% Mounted on
/dev/hda8 4294967295 0 4294967295 0% /
/dev/hda5 4016 27 3989 1% /boot
/dev/hda2 1237888 9692 1228196 1% /fun
```

La utilidad *df* se usa aquí en primer lugar sin opciones, mostrando la información sobre el uso del disco en bloques. En segundo lugar se utiliza la opción *-h* para mostrar los datos en formato legible por las personas. Por último utilizamos el comando *-i* para obtener información sobre los inodos.

Bibliografía y enlaces recomendados

LPIC 1 Certification Bible (Bible) by Angie Nash, Jason Nash
John Wiley & Sons; Bk&CD-Rom edition (July 1, 2001) ISBN: 0764547720

LPI Linux Certification in a Nutshell by Jeffrey Dean
O'Reilly & Associates; 1st ed edition (May 15, 2001) ISBN: 1565927486

CramSession's LPI General Linux Part 1 : Certification Study Guide
CramSession.com; ISBN: B000079Y0V; (August 17, 2000)

Referencias Unix Reviews
<http://www.unixreview.com/documents/s=7459/uni1038932969999/>

Página LPI: www.lpi.org

Apuntes IBM: <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

Manuales GPL: <http://www.nongnu.org/lpi-manuals/>