Preparación para el examen LPI 101

Tema 104.2

Creando particiones y sistemas de ficheros

Créditos y licencia de uso

Coordinación:
Oscar Casal (ocs) oscar@glug.es

Traducción:
Juan Maria Gil (Smooth) yo@juanmaria.com
Pere Catalan (arGos) sageta77@hotmail.com

Maquetación:
Manuel Guillán (xLekOx) lpi@xlekox.org

Versión 1.0 (23-08-2004 14:00)

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 104.2

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 trascurso 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 *directorios* 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 *directorios*.

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 */etc/fstab*. 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 *ext2*. El valor de *tipo* determina que verificador específico para el filesystem es utilizado.
-b superbloque	Utiliza una copia del *superbloque* alternativa. En el modo interactivo, e2fsck utiliza automáticamente superbloques alternativos. Normalmente, para restaurar un superbloque defectuoso, utilizarás -b 8193 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 *e2fsck*.

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

Código	***Significado***
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

Comando	***Tipo de filesystem creado***
mkfs.ext2 o mke2fs	ext2
mkfs.msdos o mkdosfs	MD-DOS
mkswap	swap
mkraid	Raid
mkfs.mimix	mimix
mkfs.bfs	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.

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. ¿Cuantos 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 _____.

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".

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.

Bibliografia 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/