

Archivado y backup

1. Las herramientas de copia de seguridad

La copia de seguridad constituye un trabajo importante del administrador, ya que, en caso de problema grave, se suele restaurar el sistema desde una copia de seguridad, o una imagen del sistema cuando éste estaba todavía bien (funcionamiento correcto, sin corrupción). Se entrega cada Unix con sus propios comandos y procedimientos de copia de seguridad; sin embargo, existen algunas herramientas comunes.

a. Comandos, planes, scripts

- ~ Para la copia de seguridad de los archivos y estructuras, utilice los comandos **tar** y **cpio**. Estos comandos guardan una estructura, y no un sistema de archivos. Se puede hacer coincidir los dos.
- ~ Para la copia de seguridad física de discos y sistemas de archivos (los dumps), utilice el comando **dd**.

Una copia de seguridad incremental consiste en salvaguardar una primera vez la totalidad de los datos y luego únicamente los archivos modificados. También se pueden encontrar en forma de programas libres o en el mercado soluciones más avanzadas de copia de seguridad (Clonezilla solución en opensource, Networker solución comercial, por ejemplo).

A veces el administrador tendrá que definir scripts de copia de seguridad y restauración adaptados a cada caso concreto (partición del sistema, datos aplicativos...) y automatizar cuando sea posible la ejecución de estos scripts en función de la fecha, hora o carga de la máquina.

También será importante definir un plan de copia de seguridad y, para ello, hay que hacerse las preguntas correctas:

- ~ ¿Qué se debe guardar?
- ~ ¿Con qué frecuencia?
- ~ ¿Cuánto tiempo se conservarán las copias de seguridad, dónde, cuántos ejemplares?

- ˘ ¿Dónde se almacenará el historial de las copias de seguridad?
- ˘ ¿Cuál es el soporte más apropiado?
- ˘ ¿Cuánta capacidad necesita el soporte de copia de seguridad?
- ˘ ¿Cuánto tiempo se prevé para guardar un archivo, un sistema de archivos?, y ¿es razonable?
- ˘ ¿La copia de seguridad debe ser automática o manual?
- ˘ ¿Cuál es el método más apropiado para la copia de seguridad?

Como cada caso es único, este libro no puede contestar a todas estas preguntas. Las respuestas dependen del entorno de destino (producción, integración, tests, etc.). Sin embargo, no se olvide de hacer una copia de seguridad del sistema (raíz, /opt, /usr, /var, /boot, etc.) después de una instalación y antes de una modificación importante, por si acaso fuera necesario volver atrás.

b. Otros comandos

- ˘ **mt**: control de una cinta magnética.
- ˘ **touch**: pone la fecha de última modificación a la hora actual, para forzar una copia de seguridad incremental.
- ˘ **find**: selecciona los archivos que hay que salvaguardar.
- ˘ **compress** y **uncompress**: compresión y descompresión de los archivos.
- ˘ **gzip**, **gunzip**, **zcat**: compresión y descompresión en formato GnuZip.
- ˘ **xz**, **unxz**, **xzcat**: compresión y descompresión del formato .xz (derivado de lzma).
- ˘ **bzip2**, **bunzip2**, **bzcat**: compresión y descompresión del formato .bz2, más eficiente que gzip.

2. tar

El comando **tar** es simple y eficaz. Crea archivos de los archivos, incluida la estructura de archivos, en cualquier tipo de soporte, incluido en otro archivo (archivo con la extensión .tar). El archivo así creado puede extenderse a varios volúmenes: cuando la cinta o el disquete está lleno, le corresponde al usuario insertar uno nuevo. El proceso de

copia/restauración continúa.

a. Archivadores

La sintaxis es la siguiente:

```
tar cvf nombre_archivo archivo(s)
```

Por ejemplo, para colocar en un archivo **tar** el directorio Desktop:

```
$ tar cvf desktop.tar Desktop/  
Desktop/  
Desktop/fusion-icon.desktop  
Desktop/konsole.desktop  
Desktop/Support.desktop  
Desktop/Office.desktop  
Desktop/Terminal.desktop  
Desktop/MozillaFirefox.desktop  
Desktop/Printer.desktop  
Desktop/.directory  
Desktop/myComputer.desktop  
Desktop/trash.desktop  
Desktop/SuSE.desktop  
Desktop/Windows.desktop
```

Los parámetros son los siguientes:

- ✓ **c**: creación de archivo;
- ✓ **v**: modo verboso, **tar** muestra lo que hace;
- ✓ **f**: el parámetro siguiente es el nombre del archivo.

b. Listar

La sintaxis es:

```
tar tvf nombre_archivo
```

Para listar el contenido del archivo anterior:

```
$ tar tvf desktop.tar
drwx----- Alejandro/users    0 2008-04-17 09:44 Desktop/
-rw-r--r-- Alejandro/users    191 2007-10-20 20:10 Desktop/fusion-icon.desktop
-rw-r--r-- Alejandro/users   4786 2007-09-26 00:43 Desktop/konsole.desktop
-rw-r--r-- Alejandro/users    665 2008-04-08 15:14 Desktop/Support.desktop
-rw-r--r-- Alejandro/users   1051 2007-10-05 10:16 Desktop/Office.desktop
-rw-r--r-- Alejandro/users   4586 2007-12-05 11:37 Desktop/Terminal.desktop
-rw-r--r-- Alejandro/users    829 2007-10-17 12:12 Desktop/MozillaFirefox.desktop
-rw-r--r-- Alejandro/users   3952 2007-10-05 10:16 Desktop/Printer.desktop
-rw-r--r-- Alejandro/users   2053 2007-10-05 10:16 Desktop/.directory
-rw-r--r-- Alejandro/users    450 2007-10-23 11:58 Desktop/myComputer.desktop
-rw-r--r-- Alejandro/users    218 2008-02-22 08:43 Desktop/trash.desktop
-rw-r--r-- Alejandro/users    328 2008-04-08 15:14 Desktop/SuSE.desktop
-rw-r--r-- Alejandro/users    472 2008-04-17 09:44 Desktop/Windows.desktop
```

El parámetro `t` lista el contenido del archivo.

c. Restauración

Para restaurar el contenido de un archivo, la sintaxis es:

```
tar xvf nombre_archivo archivos
```

Para restaurar el archivo anterior:

```
tar xvf desktop.tar
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
```

```

Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop

```

El parámetro `x` permite la extracción del conjunto de los ficheros del archivo, o de los ficheros especificados después del nombre del archivo.

d. Otros parámetros

El comando **tar** de gnu permite gestionar los formatos de compresión directamente:

- ˆ `z`: se comprime el archivo al formato gzip.
- ˆ `Z`: se comprime el archivo al formato compress.
- ˆ `j`: se comprime el archivo al formato bzip2.
- ˆ `J`: se comprime el archivo en el formato xz.

Así, los comandos anteriores para el formato de compresión **gzip** se convierten en:

```

$ tar cvzf desktop.tar.gz Desktop/
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop

```

```

Desktop/SuSE.desktop
Desktop/Windows.desktop
$ ls -l desktop.tar*
-rw-r--r-- 1 alejandro users 30720 may  9 11:16 desktop.tar
-rw-r--r-- 1 alejandro users  7556 may  9 11:22 desktop.tar.gz

```

Observe la diferencia de tamaño. Se pueden utilizar las opciones de compresión con `c`, `t` y `x`. Observe que se comprime el archivo final, y no los archivos de manera individual. Puede ser preferible no especificar una opción de compresión si está haciendo una copia en un soporte que gestiona por sí mismo la compresión.

Si comprime su archivo para destinarlo a otros sistemas, o desea guardar una compatibilidad con los parámetros por defecto de tar, puede proceder así:

```

$ gzip -cd desktop.tar.gz | tar xvf -
Desktop/
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop

```

El parámetro `-cd` precisa a **gzip** que debe descomprimir el archivo, mientras que `-c` pasa el resultado por la salida estándar. El `-` final indica a tar que debe recuperar el flujo por la entrada estándar.

Observe que las versiones recientes de tar reconocen directamente el formato de compresión, si se emplea sin pasar las opciones específicas de estos formatos.

```
# file test.tar.xz
test.tar.xz: XZ compressed data
# tar xf test.tar.xz
# echo $?
0
```

3. cpio

El comando **cpio** hace una copia de seguridad sobre la salida estándar de los archivos cuyos nombres están insertados en la entrada estándar, por defecto la pantalla y el teclado. Por lo tanto, es posible utilizar las redirecciones. **cpio** no comprime los archivos. Le corresponde a usted hacerlo.

a. Archivar

La sintaxis general es:

```
cpio -oL
```

Los parámetros más utilizados son:

- ✓ **-O**: output, creación de la copia de seguridad en salida.
- ✓ **-L**: copia los archivos relacionados, y no los vínculos simbólicos.
- ✓ **-v**: modo verboso, información detallada.
- ✓ **-c**: copia atributos de los archivos en forma ASCII (para el intercambio entre varios SO).

Veamos cómo archivar y comprimir el directorio Desktop:

```
find Desktop -print | cpio -ocv | gzip > archive.cpio.gz
Desktop
```

```

Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks
> ls -l archive.cpio.gz
-rw-r--r-- 1 alejandro users 7377 may  9 11:33 archive.cpio.gz

```

b. Listar

La sintaxis general es:

```
cpio -it archivo
```

Los parámetros son:

- `-i`: lectura del archivo en entrada.
- `-t`: como para tar, lista el contenido del archivo.

```

$ cat archive.cpio.gz | gzip -cd | cpio -it
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop

```



```

Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop
42 blocks

```

c. Restaurar

La sintaxis general es:

```
cpio -i[umd]
```

- ✦ **-u**: restauración incondicional, con borrado de los archivos que ya existen. Por defecto no se restauran los archivos si los que están presentes en el disco tienen como fecha de última modificación la misma o posterior.
- ✦ **-m**: los archivos restaurados conservan su última fecha de modificación.
- ✦ **-d**: cpio reconstruye el árbol de directorios y subdirectorios que faltan.

Para restaurar el archivo anterior:

```

$ cat archive.cpio.gz | gzip -cd | cpio -iuvd
Desktop
Desktop/fusion-icon.desktop
Desktop/konsole.desktop
Desktop/Support.desktop
Desktop/Office.desktop
Desktop/Terminal.desktop
Desktop/MozillaFirefox.desktop
Desktop/Printer.desktop
Desktop/.directory
Desktop/myComputer.desktop
Desktop/trash.desktop
Desktop/SuSE.desktop
Desktop/Windows.desktop

```

42 blocks

4. dd

El comando **dd** (*device to device*) tiene como misión la copia física, bloque por bloque, de un archivo periférico hacia un archivo periférico. Al principio se utilizaba para la lectura y escritura en cinta magnética, pero se puede utilizar también con cualquier archivo. El comando **dd** permite realizar copias físicas de discos y sistemas de archivos.

Argumento	Función
if=archivo	Nombre del archivo a la entrada (el que se debe copiar)
of=archivo	Nombre del archivo a la salida.
bs=n	Tamaño del bloque en bytes.
count=n	Número de bloques para copiar.
skip=n	Número de bloque que hay que saltar al principio del archivo de entrada.
conv=	Conversión de la entrada.
seek=	Número de bloques que hay que saltar al principio del archivo de salida.
-s	Shell (comando de conexión) por defecto del usuario (variable SHELL). El usuario puede cambiarlo mediante el comando chsh .
-p	La contraseña del usuario. ¡Cuidado! La contraseña debe estar ya cifrada. Así, a menos que se recopie la contraseña de una cuenta genérica, usted preferirá utilizar luego el comando passwd .

La opción **conv** admite los parámetros siguientes:

- ✧ **ascii**: convertir el EBCDIC en ASCII.
- ✧ **ebcdic**: convertir el ASCII en EBCDIC.
- ✧ **block**: completar los bloques que terminan por un salto de línea con espacios, hasta alcanzar el tamaño mencionado por bs.
- ✧ **unblock**: sustituir los espacios al final de bloques (de tamaño cbs) por un salto

de línea.

- ~ `lcase`: transformar las mayúsculas en minúsculas.
- ~ `ucase`: transformar las minúsculas en mayúsculas.
- ~ `noerror`: seguir incluso después de errores de lectura.
- ~ `notrunc`: no limitar el tamaño del archivo de salida.
- ~ `sync`: completar cada bloque leído con ceros para alcanzar el tamaño ibs.

Aquí, si usa el formato de particionamiento MBR, colocará el sector de boot MBR de la partición donde está grub en un archivo. El archivo así creado se podrá utilizar con el cargador de Windows u otro para un arranque en Linux.

```
# dd if=/dev/sda1 of=boot.lnx bs=446 count=1
```

Para crear un archivo vacío con un tamaño de 1 MB:

```
$ dd if=/dev/zero of=vacio bs=1024
count=1024
1024+0 registros leídos
1024+0 registros escritos
1048576 bytes (1,0 MB) copied, 0,0199192 s, 52,6 MB/s
```