Comandos básicos

Veremos los comandos básicos para hacer backups en UNIX: `tar`, `cpio` y `dump`

Comandos `dump` y `restore`

Comandos más comunes para copias de seguridad

- comandos originales de BSD UNIX
- dependen del tipo de filesystem

Comando `dump`:

Hace copias de un sistema de archivos entero, con las siguientes características:

- Pueden ser copias multivolumen
- Puede salvar ficheros de cualquier tipo (incluido ficheros de dispositivos)
- Los permisos, propietarios y fechas de modificación son preservados
- Puede realizar copias incrementales
- También puede usarse para salvar ficheros individuales (no es lo usual)

El formato y los argumentos de `dump` dependen de la versión utilizada, pero en general es:

```
dump [-nivel] [opciones] [ficheros_a_salvar]
```

- Nivel de `dump`: entero entro 0-9:
  - 0 implica backup completo
  - mayor que 0 implica copiar sólo los ficheros nuevos o modificados desde el último backup de nivel inferior
  - `dump` guarda información sobre los backups realizados en el fichero `/etc/dumpdates` o `/var/lib/dumpdates`
- Algunas opciones:
  - `-f` especifica el dispositivo o fichero donde salvar la copia
  - `-u` actualiza el fichero `dumpdates` después de una copia correcta
  - `-a` determina automáticamente el fin de la cinta (opción por defecto)
  - `-j`, `-z` usa compresión con bzlib o zlib (sólo en algunas versiones)
- Ejemplo: backup de nivel 0 de la partición `/home`

```
# dump -0u -f /dev/st0 /home
```

- Ejemplo: backup en una máquina remota usando `ssh` como transporte

```
# export RSH=ssh
# dump -0u -f sistema_remoto:/dev/st0 /home
```

Comando `restore`:

Restaura ficheros salvados por `dump`

- Formato:

```
restore acción [opciones] [ficheros_a_recuperar]
```

- Acciones principales:
  - `r` restaura la copia completa
  - `t` muestra los contenidos de la copia
  - `x` extrae sólo los ficheros indicados
  - `i` modo interactivo
    - permite ver los ficheros de la copia
    - con `add` indicamos los ficheros a extraer y con `extract` los extraemos
    - usar `?` para ayuda
- Algunas opciones:
  - `-f` especifica el dispositivo o fichero de la copia
  - `-a` no pregunta de que volumen extraer los ficheros (lee todos los volúmenes empezando en 1)
- Ejemplo: restaurar el backup de `/dev/st0`

```
# restore -rf /dev/st0
```

- Ejemplo: restaurar el backup desde un sistema remoto

```
# export RSH=ssh
# restore -rf sistema_remoto:/dev/st0
```

- Ejemplo: restaurar sólo un fichero

```
# restore -xaf /dev/st0 fichero
```

Archivo `restoresymtable`:

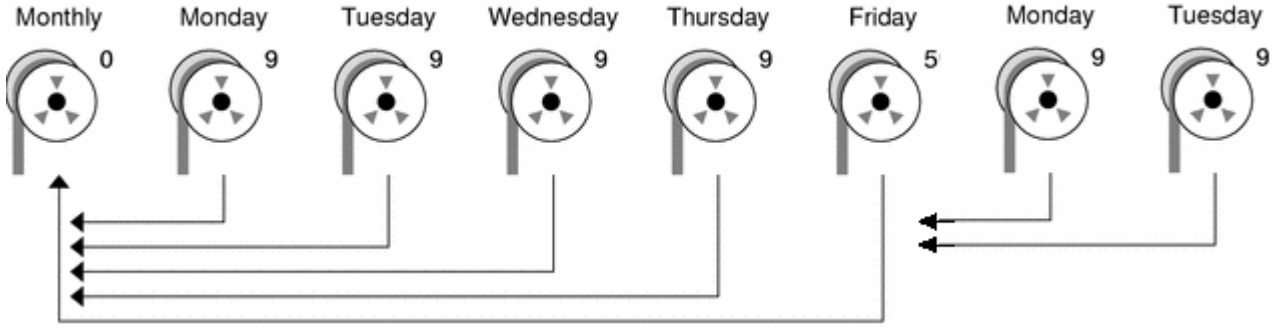
Se crea cuando se restaura un filesystem completo, en el directorio donde se restaura

- Contiene información sobre el sistema restaurado
- Puede eliminarse una vez finalizada la restauración

Planificación de los backups

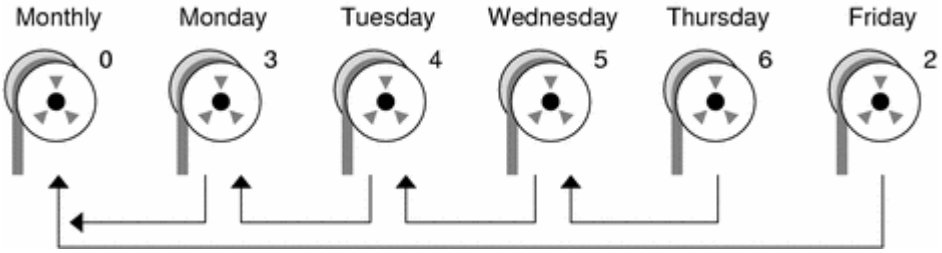
Podemos seguir diferentes estrategias a la hora de planificar los backups

- Ejemplo 1: copia de nivel 0 mensual, de nivel 9 diaria y de nivel 5 semanal



  - necesita 6 o 9 cintas: una para el 0, 4 para los niveles 5 y 1 o 4 para los niveles 9
  - para restaurar necesitamos restaurar en orden:
    1. la copia de nivel 0
    2. la última copia de nivel 5, y
    3. la última de nivel 9, después de la de nivel 5

- Ejemplo 2: copia de nivel 0 mensual, de nivel 2 semanal y de niveles 3, 4, 5 y 6 cada día



  - necesita al menos 9 cintas
  - para restaurar necesitamos restaurar en orden:
    1. la del nivel 0
    2. la del último viernes (nivel 2)
    3. las diarias desde el último viernes de forma consecutiva

Comando `tar` (Tape ARchiver)

Permite almacenar varios ficheros en uno sólo, manteniendo la estructura de directorios:

- Sintaxis:

```
tar [-]función[modificador] fichero [directorio]
```

- Ejemplo: crea un fichero tar conteniendo los ficheros del directorio `/etc`

```
tar cvf copia.tar /etc
```

- Puede indicarse un fichero o un dispositivo (p.e. `/dev/fd0`)
- `tar` conserva las propiedades de los ficheros: permisos, usuario/grupo, fechas, etc.
- Funciones principales:
  - `c` crea un nuevo archivo `tar`
  - `x` extrae los ficheros del archivo
  - `t` lista los ficheros del archivo
  - `r` añade nuevos ficheros al final del archivo `tar`
  - `u` almacena sólo los ficheros nuevos o modificados respecto a los del archivo `tar`
  - `A` añade un fichero `tar` a otro
  - `d` obtiene las diferencias entre los ficheros de la copia y los del disco
  - `--delete` borra un fichero del archivo `tar`
- Algunas opciones:
  - `v` verbose, muestra lo que está haciendo
  - `f` para indicar el nombre del fichero `tar`; por defecto toma `-` que representa la entrada/salida estándar
  - `z` comprime la copia con `gzip`
  - `--bzip2` o `j` comprime la copia con `bzip2`
  - `l` almacena sólo los ficheros locales (útil con NFS)
  - `k` no sobreescribe los ficheros existentes al extraer
  - `T` o `--files-from F` obtiene la lista de ficheros a guardar del fichero `F`
  - `X` o `--exclude-from=F` excluye los ficheros que concuerdan con los patrones listados en el fichero `F`

    - `N` o `--newer DATE` sólo guarda los ficheros más nuevos que `DATE`
    - `M` o `--multi-volume` permite crear copias multivolumen (por ejemplo, varios disquetes)
- para más opciones ver la página de info (`info tar`)
- Ejemplos:
    - Extrae todos los ficheros de `copia.tar`

        ```
        tar xvf copia.tar
        ```

    - Extrae el el fichero `passwd` de `copia.tar`

        ```
        tar xvf copia.tar etc/passwd
        ```

    - Copia el contenido de `/tmp` directamente a un disquete

        ```
        tar cvf /dev/fd0 /tmp
        ```

    - Copia un directorio completo

        ```
        (cd dir1 && tar cf - .) | (cd dir2 && tar xvf -)
        ```

    - Copia los ficheros más nuevos que un fichero `control`

        ```
        find dir -newer control ! -type d -print | tar cvfT f.tar -
        ```

- Problemas con `tar`
    - Algunas versiones no admiten opciones como la compresión o copia multivolumen
    - Algunas versiones tienen problemas con paths muy largos (más de 100 caracteres)

Comando `cpio`

El comando `cpio` es similar a `tar` en funcionalidad

- crea y extrae archivos, o copia ficheros de un lugar a otro
- maneja archivos en formato `cpio` y formato `tar`

Tres funciones primarias:

1. Copy-out: copia ficheros a un archivo, con la opción `-o`
    - Ejemplo: copia todos los directorios desde el actual en el fichero `tree.cpio`

        ```
        $ find . | cpio -ov > tree.cpio
        ```

    - para usar un dispositivo en lugar de un fichero, sustituir `tree.cpio` por `/dev/dispositivo`
2. Copy-in: extrae los ficheros de un archivo, con la opción `-i`

    ```
    $ cpio -idv < tree.cpio
    ```

    - la opción `d` crea los directorios al ir extrayendo
3. Copy-pass: usado para copiar ficheros de un árbol de directorios a otro, con la opción `-p`
    - Ejemplo: copia los ficheros del directorio actual y subdirectorios a un nuevo directorio `new-dir`

```
$ find . -depth -print0 | cpio --null -pvd new-dir
```

    - la opción `-depth` procesa primero el contenido del directorio y después el directorio (mejor para restaurar)
    - las opciones `-print0` y `--null` evitan problemas con nombres de ficheros que contengan un carácter de newline
        - `-print0` termina los nombres de los ficheros con un '\0' en vez de '\n'
        - `--null` o `-0` lee una lista de ficheros terminados por un '\0'

- Para más opciones y uso de `cpio` ver la página de información: `info cpio`

Comando `afio`

Variación de `cpio`, con varias mejoras:

- Permite hacer copias multivolumen
- Permite archivar los ficheros comprimiendolos de uno en uno
    - No comprime los ficheros que no interesa comprimir por que ya lo están (reconoce por extensión)
- Permite verificar la copia con el original (opción `-r`)

Modos de funcionamiento similares a `cpio`

- `-o` guarda a archivo, `-i` extrae de un archivo y `-p` copia directorios
- otras opciones: `-r` verifica el archivo con el filesystem; `-t` muestra el contenido del archivo
- Ejemplos:
    - Salvar a disquete multivolumen comprimido

      ```
      $ find . | afio -ov -s 1440k -F -Z /dev/fd0
      ```

    - Comprobar con el original una copia comprimida en varios disquetes

      ```
      $ afio -rv -s 1440k -F -Z /dev/fd0
      ```

    - Muestra el contenido del archivo:

      ```
      $ afio -tv -s 1440k -F -Z /dev/fd0
      ```

    - Extrae el contenido del archivo

      ```
      $ afio -iv -s 1440k -F -Z /dev/fd0
      ```

    - Copia los ficheros del directorio actual y subdirectorios a un nuevo directorio `new-dir`

      ```
      $ find . -depth -print0 | afio -p0xa directorio_nuevo
      ```

- Para opciones ver la página de manual

Comando `dd`

Comando de copia y conversión de ficheros

- Sintaxis.

```
dd [if=fichero_entrada] [of=fichero_salida] [opciones]
```

- Por defecto, copia de la entrada estándar a la salida estándar
- Algunas opciones:
  - `ibs=b` lee `b` bytes de cada vez (tamaño de bloque, por defecto 512)
  - `obs=b` escribe `b` bytes de cada vez
  - `bs=b` lee y escribe `b` bytes de cada vez
  - `cbs=b` especifica el tamaño del bloque de conversión
  - `skip=n` salta `n` bloques del fichero de entrada antes de la copia
  - `seek=n` salta `n` bloques del fichero de salida antes de la copia
  - `count=n` copia sólo `n` bloques del fichero de entrada
  - `conv=conversión` convierte el formato del fichero de entrada según el valor de `conversión`:
    - `ascii` Convierte EBCDIC a ASCII
    - `ebcdic` Convierte ASCII a EBCDIC
    - `swab` Intercambia cada par de bytes de la entrada
    - `lcase` Cambia las letras mayúsculas a minúsculas
    - `ucase` Cambia las letras minúsculas a mayúsculas
    - `noerror` Continúa después de producirse errores de lectura
- Ejemplo: imagen de floppy de 3.5, con 18 sectores por pista, dos cabezas y 80 cilindros:

```
$ dd bs=2x80x18b if=/dev/fd0 of=/tmp/floppy.image
```

  - la `b` representan bloques de 512 bytes (en total 1474560 bytes)
  - la copia se realiza de una sola vez
- Ejemplo: extrae los datos de una cinta con error

```
$ dd conv=noerror if=/dev/st0 of=/tmp/bad.tape.image
```

- Ejemplo: `tar` del directorio actual y copia en cinta en el sistema remoto

```
$ tar cjf - . | ssh remoto dd of=/dev/st0
```

Comando `mt`

Permite la manipulación directa de la unidad de cinta

- Sintaxis.

```
mt [-f unidad_de_cinta] operación [número]
```

- con `-f` indicamos la unidad de cinta a utilizar (si se omite se toma la definida en la variable `TAPE`)
- Algunas operaciones:
  - `stat(us)` muestra el estado de la unidad de cinta
  - `rew(ind)` rebobina la cinta hasta el principio

- `ret(ension)` alisa y da tensión a la cinta (rebobina hasta el principio, luego hasta el final y nuevamente al principio)
- `erase` borra la cinta entera
- `fsf`/`bsf` se avanza/retrocede el número de archivos especificado por `número`
- `eom` salta hasta el final de parte grabada