TAPER: Gestión de Copias de Seguridad

Administración de Sistemas y
Servicios Telemáticos

Miguel Martínez Parra

1. Objetivos y descripción del servicio

Taper es un programa para realizar backups sobre cintas y otros dispositivos externos. También soporta almacenamiento en disco duro y además presenta una interfaz gráfica.

La principal diferencia con otras utilidades de este tipo (por ejemplo *tar* o *cpio*) es que ofrece información "inmediata" sobre los ficheros incluidos en cada archivo (en cada archivo se almacenan varios ficheros), evitando tiempos de espera considerables (unos 5 minutos para archivos de 250 MB) debidos a la búsqueda de los ficheros citados a lo largo de todo el archivo.

Otra diferencia es la posibilidad de añadir ficheros a archivos existentes en discos extraibles.

2. Requisitos

Para el funcionamiento de **taper** se necesitan los siguientes componentes:

- Librería ***ncurses*** en una versión superior o igual a la 1.9.6.
- El ***núcleo*** de Linux debe ser por lo menos el 1.2.5. Debe soportar *SYS_V_IPC* si se quiere utilizar la opción de *triple buffer*.
- Versión del compilador ***gcc*** mayor o igual a 2.6.1.
- **Drivers** para los dispositivos de cinta: pueden ser *ftape* o *zftape* (preferible ftape versión >= 3.xx, ya que es una "fusión" de los dos). Se deben indicar dos dispositivos: uno de rebobinado automático, y otro no automático (en caso de que se utilice el disco duro como dispositivo de almacenamiento, indicar el nombre del archivo para los dos dispositivos).

No hay limitaciones del sistema: el programa ha sido adaptado para permitir la compatibilidad entre diferentes máquinas. Los archivos creados están en formato *little endian*, pero **taper** se encarga de pasarlo al formato acorde a la máquina.

3. Arquitectura del servicio

Para tener acceso directo al nombre de los ficheros almacenados, **taper** crea un archivo de información, compuesto a su vez por otros cuatro archivos: dos *principales*, y dos de *índices*.

El citado "archivo" de información (recordar que son cuatro), puede ser reconstruido en caso de que se borre o dañe, o simplemente que no esté disponible en otra máquina en la que no se hizo el backup original. Para evitar esta "reconstrucción" (como se comentó anteriormente, puede llevar bastante tiempo), podemos llevar junto a la cinta del backup, un disquete con los dos archivos *principales* (tendremos que reindexar), o los cuatro (los anteriores más los dos de *índices*), en cuyo caso la obtención de información será inmediata.

Estos cuatro archivos pueden ser muy grandes, pero el grado de compresión es de 10:1, por lo tanto es aconsejable activar la opción de compresión para el archivo de información.

Cada archivo lleva como identificador la hora del sistema en que fue creado, por lo que quedan identificados unívocamente.

El programa soporta tres tipos de compresión: *externa* (mediante *gzip*), *interna* (opción por defecto) y *gzip interna*. La segunda opción es la más rápida, aunque es la que menos comprime y la que más memoria consume. La mayor compresión se obtiene con la tercera opción, aunque el archivo resultante no se podrá descomprimir mediante *gzip* (habrá que utilizar **taper**).

4. Configuración

Para la instalación se deben seguir los siguientes pasos:

- Descomprimir los fuentes: *tar xzf taper-x.y.tar.gz.*
- Modificar *Makefile.common* según las opciones deseadas:
  - *TRIPLE_BUFFER*: mejora el rendimiento en dispositivos lentos, aunque puede ralentizar otros más rápidos. Para poder utilizar esta opción, se debe compilar el kernel con la opción *SYS_V_IPC* activada.
  - *TAPER_BIG_ENDIAN*: para máquinas que utilicen formato *big endian*.
  - *FIFO_PROBLEMS*: si se utiliza *libc5* y se están obteniendo errores.
  - *GLIBC*: si se utiliza *glibc* (*libc6*).
- Compilar: *make clean* ; *make all*.
- Instalar: *make install*. Posteriormente se puede desinstalar mediante *make uninstall*.

Es posible editar a mano *defaults.h* para una configuración más personalizada, aunque no es recomendable.

5. Pruebas

Arrancar el programa mediante **taper -T *opción***. Esa *opción* puede ser, según el dispositivo a utilizar: *ftape*, *zftape* (también para las versiones *ftape* >=3.xx), *scsi*, *ide*, *file* (archivo en disco duro) y *removable* (para disquetes o unidades ZIP).

Hay un par de utilidades que pueden mejorar considerablemente la velocidad del proceso: *quick fast forward* (avance rápido) y *seek* (búsqueda). No todos los dispositivos soportan estos modos de funcionamiento: el primero es soportado por unidades *zftape* y *SCSI*, y el segundo por *zftape*, *SCSI* e *IDE*. En cualquier caso, se puede comprobar si nuestra unidad soporta alguno de estos modos ejecutando ***Test fast fsf*** y ***Test can seek*** respectivamente dentro del menú de utilidades (***Utilities***). IMPORTANTE: estas pruebas borrarán cualquier dato almacenado en la unidad.

Antes de utilizar cualquier cinta o disco, es posible que necesite ser formateado y preparado para su uso con **taper**.

En el caso de cintas, el formateado se deberá realizar desde DOS/Windows, ya que no existen utilidades para Linux (por lo tanto es preferible comprarlas ya formateadas). Esto no será necesario en el resto de unidades.

Algunas unidades pueden necesitar ser preparadas para su uso con **taper**. Para comprobar si este es el caso, basta con ejecutar ***Test make tape*** en el menú ***Utilities***. Si el resultado es positivo, habrá que ejecutar ***Mktape*** con cada nueva unidad que utilicemos.

Es posible que nuestro sistema Linux tenga un sistema de archivos */proc*. Como contiene información del sistema en sí, no nos interesa que **taper** haga copia de estos archivos. Para saber en qué dispositivo está */proc*, podemos ejecutar ***which_device*** del menú ***Utilities***. Así podremos indicar a **taper**, mediante la opción *–j*, qué dispositivo es */proc* para que lo ignore.

En caso de que observemos problemas de memoria (sistema muy sobrecargado o con poca memoria), deberemos desactivar la opción de *triple buffer* y seleccionar otro sistema de compresión distinto al activado por defecto (una buena elección es el tercer método de compresión explicado anteriormente).

Como se comentó anteriormente, si se obtienen errores (*segmentation fault*, *broken pipe*, ...) y además se está utilizando *libc5*, se deberá activar *FIFO_PROBLEMS* en *Makefile.common*.

6. Gestión diaria

Para comenzar se selecciona la opción de ***backup*** del menú. Aparecen cuatro ventanas: la superior izquierda presenta el disco duro; la superior derecha, los ficheros incluidos en el archivo; la inferior izquierda los ficheros seleccionados; finalmente, la inferior derecha, los ficheros excluidos.

La opción por defecto sólo guarda los ficheros nuevos o con fecha posterior a la del fichero existente. Se le puede indicar que realice backup de los ficheros independientemente de si existían anteriormente o no.

Para restaurar un archivo se selecciona la opción de ***restore*** del menú. Aparecen tres ventanas: la superior izquierda muestra los ficheros incluidos en el archivo; la superior derecha muestra cómo fue construido el archivo (qué archivos se guardaron y cuáles fueron excluidos); la inferior indica los ficheros seleccionados para restaurar.

Si no encuentra el archivo de información, presenta como opción restaurar todo los ficheros (antes de darnos esta opción, tendrá que haber regenerado el archivo de información, por lo que habrá tardado un rato). También se pueden restaurar ficheros independientes (seleccionados) o los ficheros de los volúmenes más recientes (un volumen es equivalente a una sesión de backup: podemos guardar el mismo archivo en diferentes ocasiones para ver, por ejemplo, las modificaciones del mismo).

Muchas de las utilidades comentadas anteriormente se pueden realizar mediante diferentes comandos, todos accesibles desde el menú principal: regenerar el archivo de información, chequear un archivo, reindexar un archivo, detenerse frente a errores (de alimentación, ...), borrar los últimos *n* volúmenes, recuperar ficheros de archivos dañados, ... El programa puede generar *logs* conteniendo la información de los errores encontrados, inconsistencias de los archivos, ...

También es posible realizar backups automáticamente, sin estar "presente". Se pueden añadir los ficheros a un archivo existente, sobreescribir el archivo, borrar archivos con contenido "extraño", o abortar un backup si el archivo sobre el que se va a realizar el mismo no es el que hemos seleccionado inicialmente.

Para ejecutar un backup regular se puede indicar en *cron*. Hay que tener en cuenta que al no estar presente el usuario, no se piden confirmaciones, y si el dispositivo se queda sin capacidad, no podrá sustituirlo, deteniendo el proceso. Opcionalmente puede enviar un *mail* al *root* (configurable en *defaults.h*).

Si queremos realizar backups de los mismos ficheros y directorios siempre, podemos crear un *file set*: sólo tendremos que indicárselo al programa, y realizará automáticamente el proceso. El *file set* contiene información de los directorios y ficheros que queremos guardar regularmente.

Podemos utilizar variables de entorno para simplificar el trabajo. Las más importantes son:

- TAPE_TYPE: tipo de dispositivo (zftape, SCSI, ...).
- TAPER_PREFS: nombre del archivo de preferencias.
- TAPER_LOG_FILE: nombre del archivo de logs.
- TAPE: el nombre del dispositivo de rebobinado automático.
- NTAPE: el nombre del dispositivo de rebobinado no automático.

El programa admite muchas opciones de configuración mediante la inserción de opciones en la línea de comandos (al ejecutar **taper**). Esto puede resultar algo pesado, así que podremos crear un fichero de preferencias con parejas del tipo *opción=valor*. Se puede obtener una lista con las principales opciones mediante **taper –help**.

Un método para "evitar" restauraciones no deseadas de archivos es asignar al archivo de información los permisos rw-------. En cualquier caso, se debe tener en cuenta que cualquier usuario puede reconstruir este archivo y acceder al contenido de nuestro backup, por lo que la medida es más bien desalentadora, no segura.