



ULPGC
Universidad de
Las Palmas de
Gran Canaria

eii

ESCUELA DE
INGENIERÍA INFORMÁTICA

Administración de Sistemas Operativos

Tema 4.2. copias de seguridad.

Guía práctica en Linux

© 2020 José Miguel Santos

Plan de copias de seguridad en Linux

- Diseñar el plan
 - Qué archivos o directorios copiamos
 - En qué momentos
 - Qué modelo de copia: completa, incremental, diferencial...
- Acciones
 - Empaquetar archivos: **tar c**, **cpio**
 - Comprimir: **gzip**, **zip**
 - Localizar los archivos para copiar:
 - **find -newer ... | tar -T -**
 - **find -mtime ...**
 - **tar --newer *fecha***
 - **tar cf *fich.tar dir1 dir2...***
 - Planificar copias: **crontab**
 - Recuperación: **tar x**, **cpio**
 - Copias en bajo nivel: **dd**

Empaquetar archivos

- Copia de seguridad sin empaquetar:
`cp -a f1 f2 f3 ... dir_backup`
- **Problema:** se pueden perder metadatos originales (permisos, fechas)
- **Problema:** muchos sistemas de ficheros tienen un número máximo de archivos. Hay riesgo de sobrepasar este límite.
- **Solución:** empaquetar los ficheros copiados en un único archivo de copia de seguridad.

Empaquetar archivos: **tar**

- Crear un archivo TAR:

tar cf backup.tar f1 f2 f3 ...

- El fichero .tar contendrá empaquetados todos los archivos con sus permisos y demás metadatos.
- Si algún argumento es un directorio, lo copia recursivamente (se puede inhibir con --no-recursion)

- Ver la lista de ficheros del archivo TAR:

tar tf backup.tar

- Recuperación de ficheros:

tar xf backup.tar fx fy ...

- Se pueden recuperar selectivamente los ficheros que deseemos (no se obliga a recuperación completa).

Copias comprimidas

- comprimir: **gzip fichero.tar**
 - genera el archivo comprimido “fichero.tar.gz”
 - ojo, elimina el fichero original
- descomprimir: **gunzip fichero.tar.gz**
- **tar** tiene opciones para comprimir y descomprimir sin necesidad de **gzip**:
tar czf fichero.tar.gz f1 f2 f3 ...
tar xzf fichero.tar.gz fx fy ...
- También se usa la extensión **.tgz**, ej. fichero.tgz

Lista de ficheros para copiar

- como argumentos:

```
tar cf bckp.tar f1 f2 f3 ...
```

- en un fichero con la lista:

- un fichero de texto, una línea por cada ruta a copiar

```
tar cf bckp.tar -T lista
```

- por la entrada estándar:

- ponemos un "-" donde iría la ruta del fichero

```
find / -user pepe | tar cf bckp.tar -T -
```

Copia recursiva (o no)

- **tar** por defecto copia recursivamente
- Si una de las rutas es un directorio, lo copia entero y *no comprueba duplicados*:

```
tar cf A.tar /home /home/f.txt
```

/home/f.txt aparecerá dos veces dentro de A.tar

- Si queremos evitarlo: `--no-recursion`
- `find /home -user pepe |`
`tar cf bckp.tar -T - --no-recursion`

Combinar con find

- **find ... | tar cf backup.tar -T -**
 - Problema: no reconoce nombres de ficheros que empiezan por guion "-" o que contienen saltos de línea
- **find ... -print0 | tar cf bckp.tar --null -T -**
 - Solución sin efectos adversos, aunque es más complicada de entender (la lista de rutas usa el carácter NUL como separador en lugar de saltos de línea)
- Otras combinaciones:
 - tar cf bckp.tar \$(find ...)**
 - problema: no entiende las rutas que contienen espacios y caracteres especiales
 - find ... -exec tar rf bckp.tar {} +**
 - find ... -exec tar rf bckp.tar {} \;**
 - problema: se llama una y otra vez a tar, es muy ineficiente si son muchos archivos para añadir
 - find ... | xargs tar rf bckp.tar**
 - mismo problema que el **find -exec ... \;**

Copias incrementales

- Objetivo: seleccionar ficheros modificados con posterioridad a una copia anterior
- Algunos mecanismos:
`find ... -mtime -3 | tar cf ...`
`find ... -newer fichero | tar cf ...`
`tar cf backp.tar --newer 2019-06-22`
- ¿Qué fecha tomamos como referencia?
 - La fecha del fichero de copia → **OJO:** esa es la fecha de cuando *acabó* la copia, ¡¡¡ puede haber ficheros modificados durante la copia !!!
 - **Solución:** un fichero “testigo” que creamos al inicio de la copia.
Ejemplo:
 - Al hacer una copia: `touch testigo; tar cf ...`
 - Al hacer una copia incremental: `find ... -newer testigo | tar ...`

Planificar copias: **crontab**

- Se edita con **crontab -e**. Se visualiza con **crontab -l**
- Fichero crontab: líneas de texto.
Formato de la línea: *Min Hor Dia Mes Sem orden*
 - *Min*: minuto
 - *Hor*: hora (0-23)
 - *Dia*: día del mes (1-31)
 - *Mes*: número del mes (1-12)
 - *Sem*: día de la semana (0=domingo; 1=lunes; 2=martes...)
 - *orden*: Una orden o un *script* que ejecutará el shell automáticamente
- Un asterisco (*) significa «cualquier momento».
- admite rangos (ej. 1-5) y listas de valores (ej. 0,15,45)
- */5 significa «cada cinco unidades»

Ejemplos de **crontab**

- cada cuarto de hora, de lunes a viernes
`0,15,30,45 * * * 1-5 script`
- los domingos a las tres de la mañana
`0 3 * * 0 script`
- cada dos horas, los días 1 y 5 de cada mes
`0 */2 1,5 * * script`
- de enero a junio, los martes en horas impares
`0 1-23/2 * 1-6 2 script`

Ejemplo de plan con crontab

Fichero crontab

```
30 02 * * 0 /mi/script.sh 0
00 05 * * 1-5 /mi/script.sh 1
```

Hacemos una copia completa cada domingo a las 2:30AM.

De lunes a viernes a las 5AM hacemos una copia diferencial respecto a la copia completa.

Para saber qué copiamos en la diferencial usamos un fichero de testigo de fecha.

Script “script.sh”

```
#!/bin/bash
BACKUP_DIR=/mis/backups
DATE=$(date +%Y%m%d)
case $1 in
    # copia de nivel 0 (completa)
    0) touch $BACKUP_DIR/n0
        find /home |
        tar czf "$BACKUP_DIR/$DATE-0.tgz" -T -
        ;;
    # copia diferencial
    1) find /home -newer $BACKUP_DIR/n0 |
        tar czf "$BACKUP_DIR/$DATE-1.tgz" -T -
        ;;
    *) echo "Error en el script" >&2
        exit 1
        ;;
esac
```

cpio

- Utilidad equivalente a **tar**
- Diferencias con **tar**:
 - Lee la lista de archivos de la entrada estándar
 - Devuelve la copia por la salida estándar
 - No copia recursivamente directorios
 - Sirve también para hacer duplicados de directorios
- cpio se adapta muy bien a integrarse como una etapa dentro de una tubería Unix

cpio: ejemplos de uso

- **cpio -o <lista >backup.cpio**
crea un archivo a partir de una lista
- **find /home -depth | cpio -o | gzip >backup.cpio.gz**
hace una copia de /home y la comprime con *gzip*
- **cpio -id <backup.cpio**
extrae todos los archivos de la copia
- **cpio -id f1 f2 ... <backup.cpio**
extrae los archivos seleccionados
- **cpio -it <backup.cpio**
lista el contenido de una copia
- **find /home -depth | cpio -pduv carpeta.con.duplicado**
hace un duplicado de todo el /home a partir de un *find*