DAR differential backup mini-howto -ES-

Author: Grzegorz Adam Hankiewicz
Contact: dar@gradha.imap.cc
Translator: Grzegorz Adam Hankiewicz

Date: 2012-12-19
Web site: http://gradha.github.com/dar-differential-backup-mini-howto/

Translations: De la página web puede obtener este documento en inglés, italiano y español.

Indice

Introducción

Todos deberíamos hacer copias de seguridad de nuestros datos importantes. Este consejo omnipresente es habitualmente ignorado por la mayoría de las personas. Yo lo ignoré también, hasta que perdí una buena cantidad de datos importantes. Insatisfecho, continué perdiendo datos en algunos incidentes posteriores, hasta que decidí que era bastante. Entonces busqué programas de copias de seguridad en Freshmeat que permitiesen hacer copias de seguridad diferenciales y encontré DAR.

Una copia de seguridad completa significa que todos los ficheros bajo su política de seguridad serán guardados. Una copia de seguridad diferencial o incremental, sólo contendrá aquellos ficheros cuyos contenidos han cambiado desde la copia de seguridad anterior, ya sea esta completa o diferencial.

DAR le permite crear de forma sencilla un conjunto de copias de seguridad diferenciales. El método que he desarrollado me ayuda a tener copias de seguridad automáticas que se ejecutan cada noche. El primer día del mes, se realiza una copia de seguridad completa. El resto del mes, sólo se realizan copias de seguridad diferenciales. En mi situación, muy pocos ficheros cambian de un día a otro, algunas veces el código fuente del proyecto en el que estoy trabajando, y siempre mis buzones de correo.

El resultado es que puedo recuperar el contenido de mi ordenador a un día específico con facilidad, en caso de necesitarlo. DAR es un programa de línea de comando, y puede hacerse ligeramente complejo con algunas opciones. Este pequeño mini-howto le explicará mi solución personal, que es muy cruda, pero me da buenos resultados. Si, he verificado que puedo recuperar datos de las copias de seguridad. De hecho, a finales del año 2003 me trasladé a otro país y solamente llevé conmigo un CD ROM con una Knoppix autoarrancable, y recuperé el estado exacto de mi instalación Debian en cuestión de horas. Sin personalizaciones, sin largas instalaciones, sin ficheros perdidos.

Este documento fue escrito usando la versión 1.3.0 de DAR. Cuando me actualicé a DAR 2.0.3, todo seguía funcionando, ni si quiera tuve que actualizar mis archivos de copias de seguridad. Así que parece que la interfaz y el formato de copias de seguridad son bastante estables, o al menos compatibles hacia atrás. No obstante, no confíe a ciegas en este documento. Verifique que la versión de DAR que tiene instalada funciona como espera y que puede recuperar una copia de seguridad generada antes de tener que depender de ella.

Esta versión del texto usa reStructuredText (para eso son las marcas extrañas en la versión en modo texto). Lea más sobre esto en http://docutils.sourceforge.net/.

Uso simple de DAR

DAR es muy similar a tar en el número de opciones que tiene: hay suficiente para cada necesidad, pero demasiadas para un novato. Como es habitual, siempre puede obtener ayuda del programa tecleando `dar -h` o `man dar` tras su instalación. Al igual que tar, hay un conjunto de parámetros obligatorios que definen el tipo de operación que va a realizar (crear, extraer, listar, etc), y un conjunto de parámetros que afectan la opción seleccionada. Simplemente por probar, imagínese que quiere realizar una copia de seguridad de su directorio home. Escribiría algo así:

```
dar -c fichero_sin_extension -g file1 -g file2 ... -g fileN
```

La salida debería ser similar a esto:

```
$ dar -c mi_copia -g safecopy.py/ -g translate_chars.py/


 ----------------------------------------
 15 inode(s) saved
 with 0 hard link(s) recorded
 0 inode(s) not saved (no file change)
 0 inode(s) failed to save (filesystem error)
 4 files(s) ignored (excluded by filters)
 0 files(s) recorded as deleted from reference backup
 ----------------------------------------
 Total number of file considered: 19
$ ls
mailbox_date_trimmer/  mi_copia.1.dar        sdb.py/
mailbox_reader/        safecopy.py/          translate_chars.py/
```

Tal y como se habrá dado cuenta, DAR añade un número y extensión a su nombre. El propósito de la extensión es claro, ayuda a saber visualmente que el fichero es una copia de seguridad de DAR. El número es un *trozo*, y está relacionada con la característica de DAR de repartir la copia de seguridad en varios dispositivos de almacenamiento. Si por ejemplo quisiese hacer una copia de seguridad en CD ROM, pero sus directorios son mayores que la capacidad de uno, puede decirle a DAR que reparta el archivo en tantos ficheros como sea necesario, que luego puede grabar en varios CD ROMs.

¿Quiere recuperar su copia de seguridad? Muy sencillo, teclee lo siguiente:

```
$ mkdir temp
$ cd temp
$ dar -x ../mi_copia
file ownership will not be restored as dar is not run as root.
to avoid this message use -O option [return = OK | esc = cancel]
Continuing...


 ----------------------------------------
 15 file(s) restored
 0 file(s) not restored (not saved in archive)
 0 file(s) ignored (excluded by filters)
 0 file(s) less recent than the one on filesystem
 0 file(s) failed to restore (filesystem error)
 0 file(s) deleted
 ----------------------------------------
 Total number of file considered: 15
$ ls
safecopy.py/  translate_chars.py/
```

La estrategia de copias de seguridad

El primer paso para crear una buena copia de seguridad es determinar qué partes de su sistema necesitan una. Esto no significa necesariamente que no puede crear una copia de seguridad completa, sólo que repartir la copia en al menos dos partes puede ayudar mucho a DAR (y cualquier otra herramienta de copias de seguridad).

Mi sistema en casa se compone de dos discos duros. El primero está partido en una partición de 3.8 GB donde vive mi sistema completo, y otra partición de 11 GB donde almaceno mi música y otros ficheros temporales, como un repositorio local de paquetes Debian que hago para mí mismo. El segundo disco duro tiene una partición de 9.4 GB cuyo único propósito es servir de copia de seguridad del disco primario. No tengo interés en realizar copias de seguridad de mi música, porque tengo todos los CDs originales y scripts para recomprimirlos en formato ogg.

De las 3.8 GB que quiero hacer copia de seguridad, normalmente entre 1.3 y 1.5 GB están vacías. Repartiré las 2.3 GB usadas a nivel lógico entre directorios de *sistema* y *home* (en el momento de escribir esto, mi home ocupa 588 MB). La razón de esta separación es que como usuario normal sólo puedo cambiar cosas en mi directorio home y otros ficheros de las particiones que no hago copias de seguridad. Mientras, la parte *sistema* de la partición es bastante estable y no se modifica porque (des)instalo software muy de vez en cuando. De hecho, de mi directorio *home* las únicas cosas que cambian normalmente son mis directorios `Mail` y `projects`, donde pongo este documento y otro software que escribo/hackeo.

La diferenciación básica entre *directorios home* y *de sistema* también puede ser útil en organizaciones. Si trabaja para una universidad, normalmente todas las máquinas tendrán la misma configuración de sistema, pero dependiendo de la máquina sus directorios home contendrán datos diferentes. Puede hacer un a *copia de seguridad de sistema* de una sola máquina, y *copias de seguridad del home* de cada máquina. Otra configuración común es tener un servidor central que exporta los directorios home por NFS. Aquí sólo tiene que hacer copia de seguridad del servidor. Si tiene usuarios con privilegios altos, déjeles la tarea de hacer una *copia de seguridad de sistema* de sus propias máquinas, el directorio home exportado es algo que pueden ignorar dado que será realizado en el servidor.

Una vez haya decidido qué quiere guardar en su copia de seguridad, debe decidir cómo configurar DAR. Puede usar parámetros o ficheros de configuración. Los parámetros están bien cuando no tiene muchas opciones. Los ficheros de configuración son mejores cuando quiere añadir complejas reglas de inclusión/exclusión de ficheros, y además, puede usar comentarios para documentar los parámetros, indicando por ejemplo la razón por la que incluye tal o cual directorio. Esto puede ser útil si vuelve dentro de unos meses y se pregunta qué hacen todas estas opciones.

Con mi configuración, ejecutaré comandos DAR desde scripts shell llamados periódicamente por cron (Configurando algunos scripts para automatizar el proceso), así que no me importa tener largas líneas de comando, y este mismo documento tiene doble propósito para documentar esos scripts. Si prefiere

ficheros de configuración, lea la documentación de DAR para aprender su formato y cómo usarlos.

Copia de seguridad completa con DAR

Aquí está la línea de comando completa que usaré para mi copia de seguridad de *sistema*, ejecutada como **root**. No se preocupe por el gran número de parámetros, iré describiendo su propósito uno a uno:

```
dar -m 256 -y -s 600M -D -R / -c `date -I`_data -Z "*.gz" \
   -Z "*.bz2" -Z "*.zip" -Z "*.png" -P home/gradha -P tmp \
   -P mnt -P dev/pts -P proc -P floppy -P burner -P cdrom
```

- **`-m 256`** DAR puede comprimir su copia de seguridad. La compresión se aplica a ficheros individuales, y puede ser perjudicial para pequeños ficheros. Por defecto los ficheros con 100 bytes o menos no serán comprimidos. Con el parámetro `-m` incremento este valor a 256, el cual parece funcionar mejor para esos pequeños ficheros de configuración que se almacenan en `/etc/` y `/home`. Como puede ver, esta opción es completamente opcional, básicamente para fanáticos del ajuste como yo.
- **`-y [nivel]`** Esta opción activa la compresión Bzip2 del archivo, que por defecto está desactivada. Incluso puede especificar un nivel numérico de compresión, que va de 0 (no compresión) hasta 9 (mejor compresión, procesado lento). Bzip2 por defecto usa 6, que es la mejor relación velocidad/compresión para la mayoría de los ficheros. Yo no uso nivel de compresión, el 6 me va bien.
- **`-s 600M`** Aquí está la característica de DAR de trocear. El tamaño especificado de 600 Megabytes es el tamaño máximo de fichero que DAR creará. Si su copia de seguridad es mayor, obtendrá varios ficheros de copia de seguridad, cada uno con su número de trozo antes de la extensión del fichero, para que pueda salvar cada uno en una unidad diferente de almacenamiento (disquetes, zip, CDROM, etc). Mis copias de seguridad son mucho más pequeñas que este tamaño, y mantengo este parámetro sólo por si acaso se me ocurre crear un fichero grande en mi directorio home y olvido borrarlo. Si este parámetro le resulta útil, lea también en el manual de DAR sobre el parámetro `-S`.
- **`-D`** Almacena directorios como directorios vacíos aquellos excluidos por la opción `-P` o aquellos ausentes en la línea de comando como parámetros. Esto es útil cuando recupera una copia de seguridad desde cero, para que no tenga que crear manualmente todos los directorios que fueron excluidos.
- **`-R /`** Especifica el directorio raíz para salvar o recuperar ficheros. Por defecto esto apunta al directorio de trabajo actual. Estamos

realizando una *copia de seguridad de sistema*, así que apuntará al directorio raíz.

- **`-c 'date -I'_data`** Este es uno de los parámetros obligatorios de los que hablé antes, y significa crear una copia de seguridad. Para aquellos que no entienden lo que sigue, `'date -I'` es la expansión de comillas de la shell de línea de comando. En pocas palabras, `date -I` proporcionará la fecha en formato AAAA-MM-DD. Con comillas y usado como parámetro, la salida del comando será usada como cadena del comando padre. De este modo puede crear copias de seguridad con la fecha de creación empotrada en el nombre. Si todavía no sabe de lo que hablo, intente ejecutar lo siguiente desde la línea de comando:

  ```
  echo "La fecha de hoy es 'date -I'"
  ```

- **`-Z patrón_fichero`** Usando las reglas normales de meta caracteres en ficheros puede especificar patrones de ficheros que quiere almacenar en la copia de seguridad sin compresión. Esto sólo tiene sentido si usa el parámetro `-y`. Comprimir ficheros comprimidos únicamente crea ficheros mayores y malgasta tiempo de la CPU.

- **`-P ruta_relativa`** Con este parámetro le dice a DAR qué rutas no quiere almacenar en su copia de seguridad. Aquí posiblemente quiere poner el directorio home (soy el único usuario de la máquina, hay algunos más, pero con el propósito de pruebas/sistema), directorios de sistema que no son realmente ficheros físicos como `proc`, otras unidades que pueda tener montadas bajo `mnt` (destacando la unidad donde va a poner la copia de seguridad), etc, etc. Tenga en cuenta que las rutas que especifique aquí deben ser relativas a la ruta especificada por el parámetro `-R`.

Eso no fue tan difícil. En el manual de DAR puede leer sobre más parámetros que pueda querer usar. Y aquí está la linea de comando que ejecutaré como usuario dentro de mi directorio home:

```
dar -m 256 -y -s 600M -D -R /home/gradha -c 'date -I'_data \
   -Z "*.gz" -Z "*.bz2" -Z "*.zip" -Z "*.png" \
   -P instalacion_manual -P Mail/mail_pa_leer
```

Nada nuevo bajo el sol. Como puede ver, la mayoría de la línea de comando es idéntica a la anterior, únicamente cambio el nombre de los directorios que quiero excluir con `-P` y el directorio raíz con el parámetro `-R`.

Haciendo copias de seguridad diferenciales con DAR

Un vez tenga una copia de seguridad completa puede crear una copia de seguridad diferencial. La primera copia de seguridad diferencial debe ser realizada usando la copia de seguridad completa como referencia. Las

siguientes copias de seguridad diferenciales usan la última copia de seguridad diferencial como referencia. Aquí está la línea de comando para una copia de seguridad diferencial de *sistema*:

```
dar -m 256 -y -s 600M -D -R / -c `date -I`_diff -Z "*.gz" \
   -Z "*.bz2" -Z "*.zip" -Z "*.png" -P home/gradha -P tmp \
   -P mnt -P dev/pts -P proc -P floppy -P burner -P cdrom \
   -A copia_previa
```

- **`-c `date -I`_diff`** Sólo cambio el nombre del fichero, por razones cosméticas.
- **`-A copia_previa`** Este nuevo parámetro se usa para decirle a DAR dónde puede encontrar la copia de seguridad anterior para que pueda crear una copia de seguridad diferencial en lugar de una completa. La única cosa con la que debe tener cuidado es no especificar ni trozo ni extensión en el nombre del fichero, de lo contrario DAR le realizará una pregunta interactiva en la línea de comando.

La línea de comando de usuario es exactamente igual. Aquí está:

```
dar -m 256 -y -s 600M -D -R /home/gradha -c `date -I`_diff \
   -Z "*.gz" -Z "*.bz2" -Z "*.zip" -Z "*.png" \
   -P instalacion_manual -P Mail/mail_pa_leer -A copia_previa
```

DAR tiene otra buena característica que no usamos: *catálogos*. Cuando crea una copia de seguridad con DAR, internamente contiene todos los datos más un *catálogo*. Este *catálogo* contiene información sobre qué ficheros fueron guardados, sus fechas, su tamaño comprimido, etc. Puede extraer un *catálogo* y almacenarlo por separado. ¿Para qué querría hacer esto? Para configurar copias de seguridad diferenciales por red.

Para poder crear una copia de seguridad diferencial, necesita proporcionar a DAR la copia de seguridad previa para que pueda decidir qué ficheros han cambiado. Realizar esto puede consumir mucho ancho de banda en una red. En su lugar, tras crear la copia de seguridad, puede extraer el *catálogo* y enviarlo a la máquina que realiza las copias de seguridad. La siguiente vez, puede usar este fichero con el parámetro `-A`, y funcionará como si el fichero completo estuviese ahí.

Esto también puede ser útil si usa trozos, porque el *catálogo* se crea a partir del primer y último trozo. Es mucho más cómodo usar un solo fichero con el comando de copia de seguridad en lugar de tener que llevar consigo los discos de la copia de seguridad anterior.

Configurando algunos scripts para automatizar el proceso

Tal y como se mencionó anteriormente, es hora de configurar las copias de seguridad bajo cron. Ponga el siguiente script ejecutable para copias de seguridad de *sistema* bajo `/root/dar_backup.sh`:

```
#!/bin/bash

DIR=/var/backups/system
FILE=${DIR}/`/bin/date -I`_data
# Commands
/usr/local/bin/dar -m 256 -y -s 600M -D -R / -c $FILE -Z "*.gz" \
   -Z "*.bz2" -Z "*.zip" -Z "*.png" -P home/gradha -P tmp \
   -P mnt -P dev/pts -P proc -P floppy -P burner \
   -P cdrom -P var/backups > /dev/null
/usr/local/bin/dar -t $FILE > /dev/null
/usr/bin/find $DIR -type f -exec chown .gradha \{\} \;
/usr/bin/find $DIR -type f -exec chmod 440 \{\} \;
```

Algunas cosas a destacar:

- DIR es la variable que contiene el directorio destino.
- FILE contendrá la ruta a la copia de seguridad del día.
- Uso rutas completas para los comandos porque mi cuenta root no las tiene incluidas en el entorno por defecto. Esto es un riesgo de seguridad potencial. Idealmente querría compilar DAR como root y guardar los binarios donde los cree para que nadie pueda tocarlos. Y también ejecutar Tripwire sobre ellos.
- DAR genera estadísticas tras cada ejecución. No las queremos en nuestro cron porque generarían emails innecesarios. Sólo `stdout` (la salida estándar) es redireccionada a `/dev/null`. Los errores serán mostrados y un email enviado si algo va mal.
- Los últimos dos comandos `find` son opcionales. Los uso para cambiar el propietario a un usuario normal, quien creará posteriormente las copias de seguridad. De nuevo, otro riesgo de seguridad. El usuario root debería hacer copias de seguridad como root, y los usuarios deberían realizar sus propias copias. Pero en un sistema monousuario me da igual. Si algún intruso es lo suficientemente bueno para atravesar el cortafuegos y las palabras claves de mis cuentas de usuarios para poder leer las copias de seguridad, ya la he fastidiado.

Ahora ponga el siguiente script casi idéntico para copias de seguridad diferenciales en `/root/dar_diff.sh`:

```
#!/bin/bash

DIR=/var/backups/system
FILE=${DIR}/`/bin/date -I`_diff
PREV=`/bin/ls $DIR/*.dar|/usr/bin/tail -n 1`
```

```
/usr/local/bin/dar -m 256 -y -s 600M -D -R / -c $FILE -Z "*.gz" \
   -Z "*.bz2" -Z "*.zip" -Z "*.png" -P home/gradha -P tmp -P mnt \
   -P dev/pts -P proc -P floppy -P burner -P cdrom \
   -P var/backups -A ${PREV%%.*} > /dev/null
/usr/local/bin/dar -t $FILE > /dev/null
/usr/bin/find $DIR -type f -exec chown .gradha \{\} \;
/usr/bin/find $DIR -type f -exec chmod 440 \{\} \;
```

Los únicos dos cambios son la adición del parámetro `-A` y la generación de la variable PREV con una complicada línea de comando. Veamos qué es lo que hace esta línea de comando:

- Primero el comando `ls` crea un listado de los ficheros con la extensión `.dar` en el directorio de copias de seguridad. La salida se pasa por una tubería al siguiente comando.
- Por defecto `ls` muestra los ficheros en orden alfabético. Usamos `tail` para obtener el último fichero con el parámetro `-n 1`, el cual hace que sólo se muestre la última línea.
- DAR quiere operar siempre con nombres de fichero sin número de trozo o extensión. Esto significa que si no nos deshacemos de éstas, DAR detendrá la operación para realizar una pregunta interactiva al usuario, fastidiando toda la automatización. Separamos el nombre completo del fichero con una característica de Bash llamada expansión de parámetros. Hay varios tipos de expansiones posibles, puede teclear `man bash` para verlas todas. Aquella que usa `%%` eliminará el patrón final más largo que coincida con lo que va tras `%%`. El resultado es el nombre base que queremos pasar a DAR.

Ahora sólo tenemos que poner estos dos scripts bajo cron. Esto es lo que tenemos que teclear tras `crontab -e`:

```
15 0 2-31 * * ./dar_diff.sh
15 0 1    * * ./dar_backup.sh
```

Puede informarse sobre la sintaxis con `man -S 5 crontab`. En pocas palabras, estas dos líneas le dicen a cron que ejecute los scripts 15 minutos tras medianoche. `dar_backup.sh` se ejecutará sólo el primer día del mes. El otro script se ejecutará el resto de los días.

Aquí están los scripts de copia de seguridad para sus usuarios. Son iguales, cambiando únicamente los parámetros del comando DAR y algunas rutas:

```
#!/bin/bash
# dar_backup.sh

DIR=/var/backups/gradha
FILE=${DIR}/`/bin/date -I`_data
# Commands
/usr/local/bin/dar -m 256 -y -s 600M -D -R /home/gradha -c $FILE \
   -Z "*.gz" -Z "*.bz2" -Z "*.zip" -Z "*.png" \
   -P instalacion_manual -P Mail/mail_pa_leer > /dev/null
/usr/local/bin/dar -t $FILE > /dev/null
```

```
/usr/bin/find $DIR -type f -exec chmod 400 \{\} \;

#!/bin/bash
# dar_diff.sh

DIR=/var/backups/gradha
FILE=${DIR}/`/bin/date -I`_diff
PREV=`/bin/ls $DIR/*.dar|/usr/bin/tail -n 1`
/usr/local/bin/dar -m 256 -y -s 600M -D -R /home/gradha -c $FILE \
   -Z "*.gz" -Z "*.bz2" -Z "*.zip" -Z "*.zip" \
   -P instalacion_manual -P Mail/mail_pa_leer \
   -A ${PREV%%.*} > /dev/null
/usr/local/bin/dar -t $FILE > /dev/null
/usr/bin/find $DIR -type f -exec chmod 400 \{\} \;
```

No olvide añadir las entradas crontab requeridas por su usuario apuntando a la ruta adecuada.

Recuperando su copia de seguridad desde cero

Cuando llegue el momento de recuperar su copia de seguridad, dependiendo de lo que haya guardado tendrá una copia de seguridad completa del mes más copias de seguridad diferenciales hasta la última vez que las pudo realizar. El proceso de recuperación es muy simple, es el mismo descrito en el primer capítulo (Uso simple de DAR), sólo que debe hacerlo primero con la copia de seguridad completa, y entonces con las copias de seguridad diferenciales. Esto puede ser muy aburrido, así que aquí tiene otro script que puede guardar junto con sus ficheros de copia de seguridad:

```
#!/bin/bash

if [ -n "$3" ]; then
   CMD="$1"
   INPUT="$2_data"
   FS_ROOT="$3"
   $CMD -x "$INPUT" -w -R "$FS_ROOT"
   for file in ${INPUT:0:8}*_diff*; do
      $CMD -x "${file:0:15}" -w -R "$FS_ROOT"
   done
   echo "All done."
else
   echo "Not enough parameters.

Usage: script dar_location base_full_backup directory

Where dar_location is a path to a working dar binary, base_full_backup
is a date in the format 'YYYY-MM-DD', and directory is the place where
you want to put the restored data, usually '/' when run as root."
fi
```

Este script es auto explicativo. La única cosa por la que debe preocuparse es el parámetro `-w`, que le dice a DAR que sobreescriba los ficheros que encuentre. Esto es necesario para copias de seguridad diferenciales. Oh, y ponga el script en el mismo directorio que sus ficheros de copia de seguridad. Aquí tiene un ejemplo de uso:

```
./recover.sh /usr/local/bin/dar 2003-10-01 /tmp/temp_path/
```

Pruebe ejecutar eso como un usuario normal con algunos ficheros de copias de seguridad. Puede poner el resultado en un directorio temporal, así que lo bueno es que no necesita borrar su disco duro para probarlo.

Añadiendo verificaciones a los scripts

Denis Corbin sugiere que los scripts que crean las copias de seguridad podrían verificar el código de salida del comando DAR. Para el propósito de estos scripts tan simples esto no es crítico porque el propio DAR abortará la operación con un mensaje de error, y cron informará de cualquier salida de error por email (algo que no ocurre si todo va bien).

No obstante, verificar el código de salida puede ser útil si está probando los scripts de forma interactiva y quiere saber qué comandos están siendo ejecutados:

```
#!/bin/bash

DIR=/var/backups/system
FILE=${DIR}/`/bin/date -I`_data
# Commands
if /usr/local/bin/dar -m 256 -y -s 600M -D -R / -c $FILE -Z "*.gz" \
      -Z "*.bz2" -Z "*.zip" -Z "*.png" -P home/gradha -P tmp \
      -P mnt -P dev/pts -P proc -P floppy -P burner \
      -P cdrom -P var/backups > /dev/null ; then
   if /usr/local/bin/dar -t $FILE > /dev/null ; then
      echo "Archive created and successfully tested."
   else
      echo "Archive created but test FAILED."
   fi
else
   echo "Archive creating FAILED."
fi
/usr/bin/find $DIR -type f -exec chown .gradha \{\} \;
/usr/bin/find $DIR -type f -exec chmod 440 \{\} \;
```

Puede probar esta versión fácilmente ejecutando el script y matando el proceso DAR desde otra terminal o consola con `killall dar`. Esto forzará la terminación del proceso DAR y verá que una de las ramas de error es alcanzada en el script.

Otro posible uso de la verificación del código de retorno del comando sería borrar archivos incompletos del disco duro si algo falla, ejecutar comandos externos adicionales si algo falla, o evitar verificar el archivo creado

cuando sabe que el primer comando falló. Esto último se puede hacer fácilmente concatenando los comandos de creación y verificación con `&&` en una sola línea. Esto le dice a la shell que ejecute ambos comandos como una secuencia para evitar ejecutar el segundo si el primero falla.

No obstante, si falla la corriente eléctrica durante una copia de seguridad, esta versión del script todavía dejaría a medio escribir archivos inválidos. Para prevenir esto podría mejorar el script para realizar una *verificación positiva*. Esto significa crear el fichero de copia de seguridad en un directorio temporal junto con un fichero `*.valid` si se alcanza la rama adecuada del script con éxito.

Continuando esta estrategia, otro script cron monitorizando el directorio donde se crean los ficheros temporales de copias de seguridad movería al directorio final aquellos archivos con un fichero `*.valid` correspondiente, borrando todos los demás cuya última fecha de modificación fuese mayor que una hora.

Ideas para el futuro

No voy a implementar estas pronto, porque soy muy vago, pero si usted es uno de esos hackers hiper activos, aquí tiene algunas cosas que estaría bien tener:

- Unificar tanto el script principal como el diferencial en uno, por lo que si el script se ejecuta y no hay fichero de copia de seguridad principal para el mes actual, será creado, y de lo contrario se creará uno diferencia. Útil si su máquina está apagada por alguna razón durante el día del mes que realiza la copia de seguridad no diferencial.
- Mejorar los scripts para generar una imagen CDROM diaria con cdrecord y grabarla automáticamente en un disco regrabable colocado en su máquina. Por lo que si su disco duro entero resulta dañado, todavía tiene la última copia de seguridad en un otro medio de almacenamiento. Por supuesto, esto es limitado y no puede ser automático si su copia de seguridad necesita más de un CDROM. Haga lo mismo para ZIP/JAZZ/loquesea.
- Integrar las copias de seguridad generadas con una mini distribución Knoppix autoarrancable. O cualquier otra distribución basada en disquetes que puede arrancar desde CDROM. Así tendría un CDROM de rescate con las herramientas para formatear su disco duro, y justo al lado una copia de seguridad fresca con la cual restablecer su máquina a un estado funcional.
- Sincronización de los directorios con copias de seguridad a través de Internet con máquinas remotas. Así, si su máquina acaba quemándose físicamente junto con su casa, todavía tiene copias de seguridad seguras en alguna otra parte. Podría hacerse de forma sencilla con programas como rsync funcionando por ssh como tarea del cron.
- Extraer parámetros comunes en un fichero separado e incluirlo en sus scripts usando el parámetro -B de DAR. Por ejemplo:

```
$ cat > /var/backups/system/common.dcf
-m 256 -y -s 600M -D -R / -Z "*.gz" -Z "*.bz2" -Z "*.zip" \
-Z "*.png" -P home/gradha -P tmp -P mnt -P dev/pts \
-P proc -P floppy -P burner -P cdrom -P var/backups
```

Más tarde puede usar esto en el script:

```
DIR=/var/backups/system
FILE=${DIR}/`/bin/date -I`_data
# Commands
/usr/local/bin/dar -B ${DIR}/common.dcf -c $FILE > /dev/null
/usr/local/bin/dar -t $FILE > /dev/null
/usr/bin/find $DIR -type f -exec chown .gradha \{\} \;
```

¡Que también puede reusar en la versión diferencial!

De hecho, hay personas listas que han comenzado a hacer scripts de este estilo para sí mismas y no les asusta compartirlos. Para evitar engordar este mini-howto, voy a guardarlos *tal y como son* en mi página web: https://github.com/gradha/dar-differential-backup-mini-howto/tree/master/contrib.

Sientase libre de enviarme sus propias mejoras y las añadiré al directorio. Ya sea un fichero único o un `.tar.gz` con una suite de copias de seguridad completa, por favor añada un fichero simple `.txt` que pondré al lado del fichero. Por favor use inglés en su descripción, ¡y no olvide poner su nombre y dirección de correo para que la gente pueda enviarle correcciones o mejoras!

El fin

Y esa es toda la *magia*. Si tiene problemas, algo no está claro o es incorrecto (lo cual es peor), mándeme un email. Si encuentra este documento útil y quiere traducirlo, mándeme una traducción del fichero `source.en.txt` para que pueda distribuirla junto con esta versión y otros usuarios puedan encontrar fácilmente su versión traducida. Hablando de localizar, debería ser capaz de obtener el código fuente de este documento de mi página personal (enlace al comienzo del documento).

¡Disfrute!