

Tarea 1: Backups Completos e Incrementales y recuperación de archivos

Patricio Alegre D – 2623017-9

21 de diciembre del 2011

1. Abstract

Este script permite realizar backups completos ó incrementales del directorio indicado; ó recuperar archivos y/ó carpetas desde un archivo dado como argumento.

Al realizar un backup, el archivo generado tendrá el nombre del directorio dado como argumento logrando un formato:

- Para full backup: directorio.fecha-actual.f.tarz
- Para incremental backup: directorio.fecha-actual.i.tarz

Siendo:

f: Full backup, que es el respaldo completo del directorio.

i: Incremental backup, que solo considera los cambios de la última semana.

Para recuperar los archivos ó carpetas (extensión de la tarea) se utiliza la opción r, donde el script descomprime el archivo indicado, desde el archivo de respaldo <archival.tarz> en el directorio de trabajo. Si encontrase más de un archivo ó directorio con ese nombre; se listan al usuario todos los encontrados y se muestra el comando a correr para recuperar cada uno, luego se espera que el usuario reingrese la orden señalada por el script, que en este caso será ingresar el número del archivo ó directorio indicado en la ejecución.

2. Ejecución

Sintaxis: `bci ((f | i) <dir>) | (r <archival.tarz> <file>)`

Al ejecutar el comando 'bci' se pueden ingresar las siguientes opciones para respaldar:

Full backup: `bci f <dir>`

Incremental backup: `bci i <dir>`

Donde <dir> puede ser un directorio, una ruta a un directorio ó especificar el directorio actual con ".".

Para recuperar cualquier archivo ó directorio se debe utilizar la opción r:

Restaurar: `bci r <archival.tarz> <file>`

Donde <archival.tarz> solo puede ser de extensión .tarz, ó se puede especificar una ruta a un archivo .tarz.

Mientras que <file>, es nuestro archivo ó directorio a buscar dentro de <archival.tarz>, si queremos listar todo el contenido de este fichero; podemos dejar este argumento en blanco y solo ejecutar: bci r <archival.tarz>

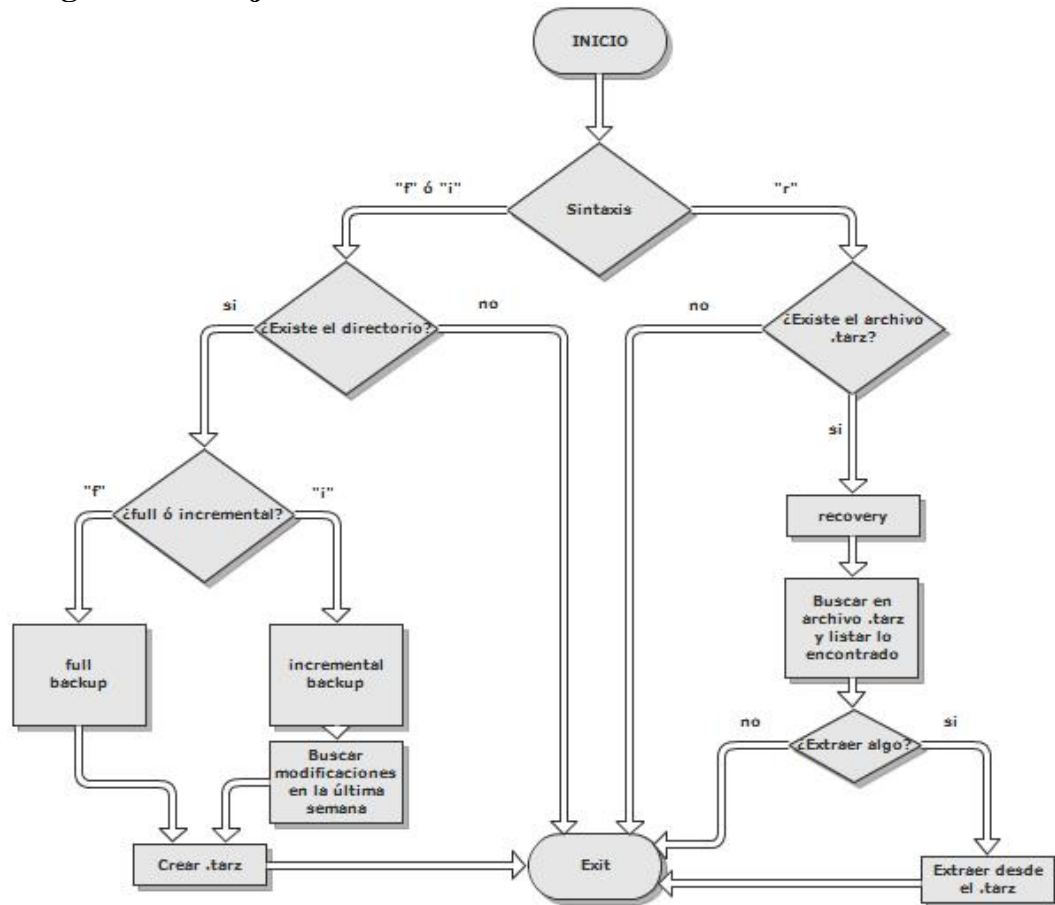
3. Funcionamiento

Primero que todo el script creado valida las opciones del usuario, es decir para la opción de backup, se certifica el ingreso de “f” ó “i” para especificarlo, luego que se indique un directorio ó ruta a uno y que este exista. Para la opción de recuperación es similar pero con el fichero ingresado, es decir se verifica que este exista, que sea del tipo .tarz y que no sea directorio.

Luego para lograr un full backup ó respaldo completo fue necesario utilizar el comando tar con las opciones de compresión, especificando el directorio a comprimir y darle el formato de salida especificado. En el caso del incremental backup se tuvo que buscar en el directorio ingresado como argumento, todos los archivos ó directorios modificados dentro de la última semana, esto se logro con find y la opción “-mtime -7”, luego a través de una pipa se enlaza esta salida al comando tar indicado anteriormente, solo que en este caso no se especifica el directorio a comprimir.

Para restaurar se creó un algoritmo que permite descomprimir más de un archivo ó carpeta si cumple el argumento de búsqueda “<file>”. Primero se listan con el comando lz el contenido dentro de “<archival.tarz>”, luego mediante un awk se busca “<file>” y se imprimen los resultados por pantalla, esta lista impresa incluye un numero identificador que nos permitirá elegir que descomprimir, a continuación el script le pide a el usuario indicar este número ó mas de uno (separados por espacio), también señala que con el número “0”, permite salir del programa. Finalmente se restaura con el comando tar, dando opciones de descompresión y especificando un directorio donde extraer, se escogió por sencillez crear un directorio en los archivos temporales con mkdir “/tmp/f\$\$”, luego se copia el contenido al directorio actual que será el directorio de trabajo, para terminar se borra lo creado en el directorio temporal.

Diagrama de flujo



4. Resultados

Listamos el contenido de un carpeta cualquiera:

```
palegred@aragorn:~/t1.sistemas$ ls -l
total 4
 1 palegred alm 1932 Dec 13 19:07 bci
 4 palegred alm  512 Dec  9 19:44 carpeta
 2 palegred alm  512 Dec 13 19:16 final
```

Podemos imprimir la sintaxis correcta solo se ingresa el comando bci:

```
palegred@aragorn:~/t1.sistemas$ bci
Sintaxis: bci ( ( f | i ) <dir> ) | ( r <archival.tarz> <file> )
```

Se realiza un backup completo de “carpeta”:

```
palegred@aragorn:~/t1.sistemas$ bci f carpeta
Escaneando el directorio: carpeta
carpeta/
carpeta/file.x
carpeta/sub carpeta/
carpeta/sub carpeta/top.txt
carpeta/subcarpeta2/
carpeta/subcarpeta2/archivo nuevo.txt
Creado el archivo: carpeta.13-12-2011.f.tarz
```

Se comprimen todos los directorios y archivos, respetando sus rutas, y se crea un archivo con el formato correcto.

Ó se realiza un incremental backup de “carpeta”:

```
palegred@aragorn:~/t1.sistemas$ bci i carpeta
Escaneando el directorio: carpeta
carpeta/
carpeta/file.x
carpeta/sub carpeta/
carpeta/subcarpeta2/
carpeta/subcarpeta2/archivo nuevo.txt
Creado el archivo: carpeta.13-12-2011.i.tarz
```

Donde podemos observar que no se comprimieron todos los archivos, solo los modificados en la última semana, luego se crea un archivo con el formato correcto.

Posibles errores de sintaxis:

```
palegred@aragorn:~/t1.sistemas$ bci f
Error, el directorio: , no existe
palegred@aragorn:~/t1.sistemas$ bci f hola
Error, el directorio: hola, no existe
```

Como ejemplo restauraremos “file.x” y “subcarpeta 2”, primero listamos todo el contenido del archivo “carpeta.13-12-2011.f.tarz”, luego los especificamos de la lista cuando el script lo solicite:

```
palegred@aragorn:~/t1.sistemas$ bci r carpeta.13-12-2011.f.tarz
Buscando ...
Reading directory of "carpeta.13-12-2011.f.tarz".
1) 0 2011-12-09 19:44:46 carpeta/
2) 0 2011-12-09 04:59:23 carpeta/file.x
3) 0 2011-12-10 23:53:15 carpeta/sub carpeta/
4) 898 2011-10-22 09:06:52 carpeta/sub carpeta/top.txt
5) 0 2011-12-13 18:57:18 carpeta/subcarpeta2/
6) 0 2011-12-13 18:57:18 carpeta/subcarpeta2/archivo nuevo.txt
6 Encontrados
Ingrese los numeros de la lista que desea descomprimir, 0 para salir
2 5

Reading directory of "carpeta.13-12-2011.f.tarz".
Copiando "file.x" al directorio actual
carpeta/file.x
Copiando "subcarpeta2" al directorio actual
carpeta/subcarpeta2/
carpeta/subcarpeta2/archivo nuevo.txt
```

Nota: la imagen es solo para ejemplificar, por lo que no se imprimen los permisos.

Verificamos listando el contenido del directorio de trabajo:

```
palegred@aragorn:~/t1.sistemas$ ls -l
total 7
1 palegred alm 1932 Dec 13 19:07 bci
4 palegred alm 512 Dec 9 19:44 carpeta
1 palegred alm 768 Dec 13 21:04 carpeta.13-12-2011.f.tarz
1 palegred alm 231 Dec 13 21:07 carpeta.13-12-2011.i.tarz
1 palegred alm 0 Dec 13 21:15 file.x
· 2 palegred alm 512 Dec 13 19:16 final
· 2 palegred alm 512 Dec 13 21:15 subcarpeta2
```

Otra variante es buscar dentro del backup, por ejemplo "top.txt"

```
palegred@aragorn:~/t1.sistemas$ bci r carpeta.13-12-2011.f.tarz top
Buscando top...
Reading directory of "carpeta.13-12-2011.f.tarz".
1)      palegred/alm      898 2011-10-22 09:06:52 carpeta/sub carpeta/top.txt
1 Encontrados
Ingrese los numeros de la lista que desea descomprimir, 0 para salir
1
Reading directory of "carpeta.13-12-2011.f.tarz".
Copiando "top.txt" al directorio actual
carpeta/sub carpeta/top.txt
```

Verificamos listando el contenido del directorio de trabajo:

```
palegred@aragorn:~/t1.sistemas$ ls -l
total 8
1 palegred alm 1932 Dec 13 19:07 bci
4 palegred alm 512 Dec 9 19:44 carpeta
1 palegred alm 768 Dec 13 21:04 carpeta.13-12-2011.f.tarz
1 palegred alm 231 Dec 13 21:07 carpeta.13-12-2011.i.tarz
1 palegred alm 0 Dec 13 21:15 file.x
2 palegred alm 512 Dec 13 19:16 final
2 palegred alm 512 Dec 13 21:15 subcarpeta2
1 palegred alm 898 Dec 13 21:24 top.txt
```

Posibles errores de sintaxis:

```
palegred@aragorn:~/t1.sistemas$ bci r carpeta.13-12-2011.f.bla top
Error el archivo: carpeta.13-12-2011.f.bla no existe ó es de otro tipo
```

5. Otros y las no-especificaciones de la tarea:

El formato de fecha en el archivo se elige: `$(date +"%d-%m-%Y")` que imprime: 08-12-2011 por ejemplo.

Para listar el contenido de un archivo .tarz, se elige trabajar con el comando `lz`, ya que entrega mucha más información que el `tar -ztf`.