DAR differential backup mini-howto -IT-

Author: Grzegorz Adam Hankiewicz
Contact: dar@gradha.imap.cc
Translator: David Gervasoni
Contact: davidgerva@gmail.com

Date: 2012-12-19
Web site: http://gradha.sdf-eu.org/textos/backup.en.html

Translations: From the web site you can get this document in English, Italian and Spanish.

Indice

Introduzione

"Chiunque dovrebbe fare le copie di backup dei suoi dati importanti". Questo avviso presente ovunque è generalmente ignorato da molta gente. Anche io l'ho ignorato, fino al giorno in cui ho perso una considerevole mole di dati. Non abbastanza contento ho fatto in modo di perderne ancora in una serie di successivi incidenti, per poi decidere che ne avevo abbastanza. Ho cercato quindi su Freshmeat qualche programma per la creazione di backup che supportasse anche la creazione di backup differenziali e ho trovato DAR.

Fare un backup completo (o base) significa salvare tutti i files che ricadono sotto le cartelle interessate dalla politica di backup. Un backup differenziale o incrementale conterrà invece solo i files il cui contenuto è cambiato rispetto al precedente backup, fosse esso completo o differenziale.

DAR permette di creare facilmente una serie di backup differenziali. Una soluzione che ho sviluppato esegue ogni notte dei backup automatici. Il primo giorno del mese viene fatto un backup completo. Il resto del mese vengono fatti solo backup differenziali. Per quanto mi riguarda i files che cambiano giornalmente non sono molti: il codice sorgente del progetto a cui sto lavorando e, più spesso, le e-mail.

Così, quando mi serve, posso recuperare con facilità il contenuto che presentava il mio computer uno specifico giorno. DAR si presenta come un programma semplice ed essenziale eseguibile da linea di comando, ma si può rendere un po' più complicato con poche opzioni. Questo piccolo mini-howto vi illustrerà la mia specifica configurazione, molto grossolana, ma, nel mio caso, funzionale. Ho già sperimentato il recupero dei dati dalle

copie di backup. Infatti verso la fine del 2003 mi sono trasferito in un altro paese e ho portato con me giusto un CD ROM e una Knoppix bootable e ho recuperato l'esatto stato della mia vecchia installazione Debian in poche ore. Senza modifiche, senza alcuna ulteriore installazione e senza perdere alcun file.

Questo documento è stato scritto usando la versione 1.3.0 di DAR. Quando sono passato alla 2.0.3 tutto funzionava. Non ho nemmeno dovuto aggiornare i miei backup. Quindi sembra che l'interfaccia e i formati di backup siano stabili o al limite compatibili con le versioni precedenti. Comunque non prendete tutto ciò che dico (scrivo) quì come garantito. Verificate prima che la versione di DAR che avete installato funzioni come dovrebbe e potrete, in futuro, recuperare i files dai backup senza problemi.

Per questa versione del testo ho usato reStructuredText (ecco spiegato il misterioso markup nella versione txt). Vedi http://docutils.sourceforge.net/ per maggiori informazioni.

Utilizzo essenziale di DAR

DAR è molto simile a tar nel numero di opzioni che ha: ce n'è una per ogni necessità, ma questo comporta una maggiore difficoltà iniziale per il nuovo utente. Come sempre, in qualsiasi momento, potete avere degli aiuti relativamente ai comandi disponibili scrivendo `dar -h` o `man dar` dopo che l'avete installato. Come nel programma tar, esistono una serie di opzioni obbligatorie che definiscono il tipo di operazione che intendete fare (creare, estrarre, listare etc) e un'ulteriore serie di opzioni che modificano la scelta prima effettuata. Giusto per esempio immaginate di voler fare un backup di una cartella della vostra directory /home. Dovrete scrivere qualcosa di simile a questo:

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

L'output dovrebbe essere simile al seguente:

```
$ dar -c my_backup_file -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 file(s) ignored (excluded by filters)
 0 file(s) recorded as deleted from reference backup
 ----------------------------------------
 Total number of file considered: 19
$ ls
mailbox_date_trimmer/  my_backup_file.1.dar  sdb.py/
mailbox_reader/        safecopy.py/          translate_chars.py/
```

Come avrete notato DAR aggiunge al nome del file un numero e un'estensione. Il motivo dell'estensione è chiaro, aiutare a capire che il file è un backup fatto con DAR. Il numero è chiamato *slice* ed è connesso alla possibilità di DAR di dividere il file di backup in base a grandezze specificate, in modo da poterle memorizzare su diversi supporti. Se per esempio voleste avere i backup su CD ROM, ma i backup delle vostre directory sono più grandi della capacità del CD ROM, potete chiedere a DAR di dividere l'archivio in tanti files che potrete poi memorizzare su diverse unità.

Volete recuperare questo backup? Scrivete semplicemente i seguenti comandi:

```
$ mkdir temp
$ cd temp
$ dar -x ../my_backup_file
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 fileystem
 0 file(s) failed to restore (fileystem error)
 0 file(s) deleted
 ----------------------------------------
 Total number of file considered: 15
$ ls
safecopy.py/  translate_chars.py/
```

La politica di backup

Il primo passo per creare backup funzionali è determinare quali parti del vostro sistema necessitano di essere archiviate. Questo non sta a significare che non potete semplicemente fare un backup del vostro intero sistema, ma dividerlo in almeno due parti aiuterà molto DAR (o qualsiasi altro tool di backup) nel suo lavoro.

Il sistema inplementato in casa mia conta di due hard disk. Il primo hard disk è diviso in una partizione da 3.8 GB, dove risiede il mio intero sistema, e un'altra partizione da 11 GB dove sono memorizzati tutta la mia musica e altri file temporanei, ad esempio alcuni pacchetti Debian fatti da me. Il secondo hard disk ha una partizione da 9.4 GB e il suo unico scopo è di servire come backup del disco primario. Non mi interessa fare il backup dei file musicali perchè ho tutti i cd originali e uno script per estrarre di nuovo le tracce e riconvertirle in ogg.

Della partizione da 3.8 GB di cui voglio fare il backup generalmente sono liberi all'incirca 1.3 - 1.5 Gb. Ho diviso "logicamente" i 2.3 GB occupati

in *system* e *home directories* (mentre scrivo, la mia home è di 588 MB). La ragione di questa divisione è che, come un normale utente, posso esclusivamente modificare il contenuto della mia home directory e alcuni file della partizione di cui non ho intenzione di fare il backup. Contemporaneamente il settore della partizione in cui risiede il sistema rimane abbastanza stabile e immutato perchè raramente (dis)installo software. Infatti anche nella mia *home* directory le sole cose che cambiano sono abitualmente la mia cartella `Mail` e `progetti`, dove metto documenti come questo e altri software che scrivo/modifico.

La distinzione di base fra *home directories* e *system* può essere anche utile nella normale organizzazione. Se lavori per una università spesso tutte le macchine hanno la stessa configurazione di base, ma ogni macchina avrà i suoi dati memorizzati. Puoi fare un singolo *system backup* di una singola macchina e più *home backup* per ogni computer. Un'altra configurazione comune è l'esistenza di un server centrale che condivide le home directories in NFS. In questo modo dovete solo fare il backup del server. Se vi sono utenti con privilegi di alto livello permettete loro di fare il backup del sistema delle loro proprie macchine, il backup delle home lo possono ignorare visto che se ne occuperà il server.

come configurare DAR. Potete usare le opzioni o i file di configurazione. Le opzioni sono utili quando non ne avete troppe da specificare. I file di configurazione sono invece meglio quando volete fare backup differenti, complessi, con inclusioni/esclusioni; inoltre potete usare commenti per documentare le opzioni specificate spiegando per esempio perchè includete/escludete questa o quella directory. Può essere utile ciò se tornate ad utilizzare il computer dopo molto tempo e volete sapere il perchè di ogni opzione.

La mia configurazione fa partire il programma DAR con una script shell richiamato periodicamente da cron (Qualche script per automatizzare i processi), così non devo digitare ogni volta lunghe stringhe di comando. Questo breve documento vuole anche introdurre brevemente alla creazione di tali scripts. Se preferite utilizzare i file di configurazione leggete la documentazione allegata a DAR per sapere come e quale sintassi utilizzare.

Eseguire backup di base (full backup) con DAR

Ecco qua sotto l'intera linea di comando che, da **root**, devo utilizzare per il backup del mio *sistema*. Non dovete preoccuparvi vedendo il gran numero di opzioni inserite, successivamente descriverò il motivo di ognuna di esse:

```
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 può comprimere i backup. La compressione è applicata a ogni file e può essere anche inutile per file di ridotte dimensioni. Di default, file di 100 bytes o meno non vengono compressi. Con l'opzione `-m` si porta questo limite a 256, cosa

che sembra funzionare meglio per tutti quei piccoli file di configurazione che stanno sotto `/etc/` e `/home`. Come potete notare questa è un'opzione assolutamente facoltativa, quasi un "capriccio".

- **`-y [level]`** Questa opzione attiva la compressione Bzip2 che di default non è attiva. Potete anche specificare un livello di compressione tramite un numero che può andare da 0 (nessuna compressione, processo veloce) a 9 (miglior compressione, processo lento). Bzip2 di default usa il livello 6 che è il rapporto migliore velocità/compressione per la maggior parte dei file. Personalmente non specifico il livello di compressione, 6 mi va più che bene.
- **`-s 600M`** Ecco quà l'opzione di DAR che vi permette di definire la dimensione dei file di backup o, meglio, delle slice. La grandezza specificata, in questo caso di 600 MB, sarà il massimo spazio occupato dai file creati. Se il vostro backup è più grande, ritroverete differenti file di backup con un numero di progressione inserito appena prima dell'estensione, cosìcchè potrete salvare ogni file su differenti supporti (floppies, zip, CDROM, etc). I miei backup sono molto più piccoli di questa dimensione e mantengo questa opzione giusto per tranquillità, nel caso i file diventassero più grandi. Se pensate che questa opzione possa esservi utile potete leggere il manuale di dar per saperne di più.
- **`-D`** Memorizza il nome e il percorso delle directory escluse dall'opzione `-P` o che non ci sono fra quelle specificate alla linea di comando. Questa è un'opzione utile quando state recuperando un backup dal nulla; in questo modo non dovete creare manualmente tutte le directory escluse.
- **`-R /`** Specifica la directory di root (directory radice) in cui salvare o dalla quale 'leggere' i file interessati dal backup. Di default questa è la directory in cui si sta lavorando (./). Se stiamo facendo un *backup di sistema* dalla cartella x, ecco che questa sarà la directory di root.
- **`-c 'date -I'_data`** Questa è l'opzione obbligatoria di cui vi ho parlato prima e definisce la creazione del backup. Per chi non capisce ciò che segue `'date -I'` è un trucchetto della shell. Brevemente, `date -I` restituisce una data con formato YYYY-MM-DD. L'output del comando fra gli apici singoli sarà usato come input dell'opzione -c. In questo modo potete creare backup con la data di creazione direttamente nel nome del file. Se ancora non capite di cosa sto parlando, provate la seguente istruzione dalla linea di comando:

  ```
  echo "La data di oggi è 'date -I'"
  ```

- **`-Z file_pattern`** Usando come argomento normali estensioni di file potete decidere quali file volete memorizzare nel vostro backup senza che siano compressi. Questo ha senso solo se usate anche l'opzione `-y`. Comprimendo file compressi otter-

rete al massimo file più grandi, nonchè spreco di risorse e occupazione della CPU.

- **`-P relative_path`** Con questa opzione dite a DAR quali directory non volete memorizzare nel vostro backup. Quì potreste mettere ad esempio la /home (Sono l'unico utilizzatore di questa macchina, ce ne sono pochi altri, ma solo per testare alcune funzioni), directory di sistema che non sono realmente dei file, come `proc`, altri file che potreste aver montati sotto `mnt` (come, ovviamente, il drive in cui metterete i file di backup) etc, etc. Notate che i percorsi che inserite devono essere relativi a quello specificato con l'opzione `-R`.

Tutto ciò non è poi così difficile. Controllate le pagine di manuale di DAR per maggiori informazioni sulle opzioni che vi interessa usare. Ed ecco quì il comando che uso all'interno della mia 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
```

Nulla di nuovo sotto il sole. Come potete vedere molti dei comandi sono identici a quelli 'di cui sopra', ho solo cambiato il nome delle directories che voglio escludere utilizzando l'opzione `-P` e la directory radice con l'opzione `-R`.

Eseguire backup differenziali con DAR

Una volta che avete creato un backup base, potete creare quelli differenziali. Il primo backup differenziale deve essere creato usando quello di base come riferimento. I backup differenziali successivi useranno come riferimento l'ultimo backup differenziale disponibile. Ecco quì il comando per un backup differenziale del *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 previous_backup
```

- **`-c `date -I`_diff`** Ho solo cambiato il nome del file, per un motivo... "pratico".
- **`-A previous_backup`** Questa nuova opzione viene usata per dire a DAR dove trova il file di backup precedente in modo da creare un backup differenziale invece di uno base. L'unica cosa alla quale fare attenzione è che voi non dovete specificare nè il numero progressivo nè l'estensione, diversamente DAR porrebbe una richiesta alla linea di comando.

La linea di comando dell'utente è esattamente la stessa. Ecco quà per completezza:

```
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 previous_backup
```

DAR ha un'altra interessante caratteristica che quì non usiamo: i *cataloghi*. Quando create un backup con DAR questo contiene i dati e un *catalogo*. Questo *catalogo* contiene informazioni inerenti i file che sono stati salvati: la loro data, la loro dimensione dopo la compressione, etc. Potete estrarre il *catalogo* e memorizzarlo separatamente. Perchè dovreste farlo? Per implementare backup differenziali in rete, ad esempio.

Al fine di creare un backup differenziale dovete procurare a DAR il backup precedente in modo che il programma possa decidere quali file sono stati modificati e quali no. Facendo questo lavoro su di una rete ciò può occupare molta banda. Invece, dopo aver creato il backup, potete estrarre il *catalogo* e inviarlo alla macchina designata alla creazione dei backup. Successivamente potete usare questo file con l'opzione `-A`, in questo modo DAR lavorerà come se il file del backup base fosse quello.

Questo può essere anche utile se usate le slices perchè il *catalogo* è creato per la prima e l'ultima slice. E' più semplice passare al comando un singolo file piuttosto che dover utilizzare tutti i dischi del vostro precedente backup.

Qualche script per automatizzare i processi

Come ho detto prima è venuto il momento di mettere la nostra procedura di backup sotto cron. Mettendo il seguente script eseguibile per il backup del *sistema* sotto `/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 \{\} \;
```

Alcune cose da notare:

- DIR è la variabile che rappresenta la directory di destinazione.
- FILE rappresenta il percorso del file di backup di oggi.
- Uso percorsi assoluti nei comandi perchè il mio account di root non li ha tutti inclusi nell'ambiente di default. Questo è potenzialmente un rischio in ambito di sicurezza. Idealmente dovreste compilare DAR

come root e mantenere i binari dove li avete creati, così nessuno potrà toccarli o eseguirvi Tripwire.

- DAR genera statistiche dopo ogni esecuzione. A noi non servono se eseguite in cron perchè produrrebbero solo mail inutili. Lo `stdout` è rediretto a `/dev/null`. Gli errori saranno invece riportati in una mail nel caso qualcosa andasse storto.
- Gli ultimi due comandi `find` sono opzionali. Li uso per cambiare i permessi dei file per un normale utente che creerà successivamente i backup. Un ulteriore rischio in fatto di sicurezza. Root dovrebbe eseguire il backup dei file da root e gli utenti i loro. Ma con un sistema mono-user questo non è importante. Se un ipotetico intruso è capace di passare attraverso il mio firewall, inserire la mia password e quindi guardare tutti i miei backup: sono fregato.

Ora ponete il seguente script per i backup differenziali, quasi identico al precedente, sotto `/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 \{\} \;
```

Gli unici due cambiamenti sono le aggiunte dell'opzione `-A` e la generazione della variabile PREV con una linea di comando un po' complicata. Vediamo cosa fa questa linea di comando:

- Prima di tutto, il comando `ls` crea una lista dei file con estensione `.dar` presenti nella directory di backup; euesto output è rediretto al comando successivo.
- Di default `ls` elenca i file in ordine alfabetico. `tail` è usato per ottenere l'ultimo file con l'opzione `-n 1` che ordina di mostrare solo l'ultima riga.
- DAR necessita di lavorare con filenames senza il numero di slice e senza estensione. Se non correggiamo noi il nome del file, DAR fermerà il processo e chiederà all'utente se effettuare l'operazione in modo automatico o meno. Separiamo quindi il nome del file con una feature Bash, chiamata parametro d'espansione. Ci sono diverse possibili espansioni, potete digitare `man bash` per vederle tutte. Usando `%%` rimuoviamo la più lunga "coda" di caratteri che si trova dopo il `%%`. Il risultato è il nome base che vogliamo passare a DAR.

Ora dobbiamo solo mettere questi due script sotto il controllo di cron. Questo è ciò che dobbiamo scrivere dopo il comando `crontab -e`:

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

Controllate in `man -S 5 crontab` la sintassi del comando. In breve queste due linee dicono a cron di far partire i processi 15 minuti dopo la mezzanotte. `dar_backup.sh` verrà eseguito solo il primo giorno del mese. L'altro script verrà eseguito tutti gli altri giorni.

Ecco quì gli scripts di backup per i vostri utenti. Essi sono identici, cambiano solo alcune opzioni di DAR e i percorsi:

```
#!/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 \{\} \;
```

Non dimenticate di aggiungere a crontab le stringhe richieste per i votri utenti.

Estrarre i backup su macchine vuote

Venuto il momento di recuperare i vostri backup, in base a quello che avete salvato, avrete il backup completo del mese e tanti backup differenziali quanti quelli che avete fatto. Il processo di recupero dei dati è molto semplice: è uguale a quello descritto nel primo paragrafo (Utilizzo essenziale di DAR), l'importante è che prima recuperiate il backup base e solo successivamente quelli differenziali. Questo può essere noioso, così ecco quà un'altro script che potete salvare fra i vostri file di backup:

```
#!/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.

Usa: script dar_location base_full_backup directory

Dove dar_location è un percorso alla directory con i binari di dar,
base_full_backup è una data in formato 'YYYY-MM-DD' e directory è
il posto dove volete mettere i file recuperati, solitamente '/'
quando eseguito come root."
fi
```

Lo script si spiega da solo. L'unica cosa alla quale dovete fare attenzione è l'opzione `-w` che dice a DAR di sovrascrivere i file trovati. Questo è obbligatorio per i backup differenziali. Ricordate di mettere lo script nella stessa directory dove mettete i file di backup. Ecco un'utilizzo di esempio:

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

Provate ad utilizzare questo come utente normale con pochi file di backup. Potete mettere i file recuperati in una directory temporanea, così non dovete svuotare il vostro hard disk per provarlo.

Aggiungere dei controlli allo script di backup

Denis Corbin suggerisce che lo script di creazione dei backup verifichi anche l'exit status dei comandi di DAR. Per quanto riguarda questo script così semplice, ciò non è di importanza critica perchè DAR stesso stamperebbe a schermo un messaggio d'errore e cron lo riporterebbe via mail (cosa che normalmente non succede se tutto va per il verso giusto)

Comunque testare l'exit status può essere utile se state verificando il funzionamento dello script e volete sapere quali comandi sono eseguiti:

```
#!/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 \{\} \;
```

Potete testare facilmente questa versione facendo partire lo script e killando i processi di DAR manualmente da un'altro terminale o un'altra console con `killall dar`, che forzerà la fine dei processi DAR e vedrete che uno dei rami di fallimento sarà raggiunto nello script di backup.

Un'ulteriore possibile utilizzo per testare il codice può essere la rimozione di archivi incompleti dall'hard disk se qualcosa andasse male o evitare di testare l'archivio creato quando sapete che il primo comando è già fallito. Successivamente si possono facilmente concatenare i comandi di creazione e di test con `&&` in una singola linea di testo. Ciò indica alla shell di eseguire entrambi i comandi in sequenza e impedisce l'esecuzione del secondo se il primo è fallito.

una procedura di backup, questa versione dello script lascierà archivi errati vaganti. Per prevenire ciò potete fare in modo che lo script esegua una *positive verification*. Ciò creerà il backup in una directory temporanea insieme con un file `*.valid`.

Così un'altro script monitora la directory dove i file temporanei sono messi e sposta in una directory definitiva i file con `*.valid` eliminando quelli la cui ultima modifica è precedente a un'ora.

Idee per il futuro

Non ho programmato di aggiornare questo testo presto perchè sono molto pigro, ma se voi siete fra quegli hackers imperattivi, ecco quà qualcosa che mi piacerebbe inserire:

- Unificare gli script dei backup di base e differenziali in uno unico, cosicchè se all'esecuzione dello script non esistono backup base per il mese corrente questo venga reato. Utile per macchine che rimangono spente molto tempo dopo che il backup mensile è stato fatto.
- Aggiornare lo script in modo che crei giornalmente un immagine per CD ROM con cdrecord e la masterizzi automaticamente su un cd riscrivibile presente nel drive. Così nel caso l'intero hard disk si guasti sarebbe disponibile l'ultimo backup su un media rimovibile. Certo la cosa è limitata e non può essere automatica nel caso i backup occupino più spazio di un CDROM. La stessa cosa vale per ZIP/JAZZ/qualsiasi cosa vogliate.

- Integrazione dei backup generati con una mini Knoppix bootable o qualsiasi altra ditribuzione che possa essere avviata da CDROM. Così avreste un CDROM per recuperare i dati che può partire automaticamente e formattare il vostro hard disk.
- Sincronizzazione delle directory di backup attraverso internet con hosts remoti. In questo modo se l'intera macchina è bruciata fisicamente, ad esempio con la vostra casa, voi avete i vostri backup in qualche altro posto. Potrebbe essere fatto facilmente con programmi come rsync attraverso ssh eseguiti tramite cron.
- Inserimento dei parametri comuni in un file separato da inculdere dallo script utilizzando l'opzione di DAR, -B. Per esempio:

```
$ 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
```

  Successivamente si può utilizzare questo nello 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 \{\} \;
```

  Che può essere riutilizzato anche nella versione differenziale!

In effetti, qualcuno ha già iniziato a creare qualche script a proprio uso e consumo e non ha problemi a condividerli. Per evitare di "disordinare" questo mini-howto ho intenzione di archiviarli *come sono* nel mio spazio web: http://gradha.sdf-eu.org/dar_scripts/.

Sentitevi liberi di inviare i vostri lavori e i vostri aggiornamenti e li aggiungerò alla directory. Se avete intenzione di inviare un singolo file di script o un `.tar.gz` con una intera suite di backup, inserite un semplice file `.txt` descrittivo che metterò assieme agli altri files, così la gente potrà leggere cosa sono e cosa fanno i files prima di scaricarli. Usate l'inglese nella vostra descrizione e non dimenticate di mettere nome e e-mail così la gente potrà inviarvi bugfixes o miglioramenti.

The end

And that's the whole *magic*. Se avete qualche problema, qualcosa non è chiaro o sbagliato (il che è peggio) inviatemi un'e-mail. Se trovi questo documento utile e lo vuoi tradurre inviami una traduzione del file `source.en.txt` così posso distribuirla assieme a questa versione e gli utenti troveranno più semplice la versione nella loro lingua. Dovreste raggiungere facilmente il codice di questo dodumento alla mia home page (link at the beginning of the document).

Enjoy!

Per finire

Versione un po' corretta, con un italiano un po' più scorrevole. Ecco lo scopo che mi ero prefissato per questa revisione. Non so se l'obbiettivo è stato raggiunto, ma non immaginavo che tradurre dall'inglese fosse così difficile. Termini che initaliano non sai come rendere, plurali di nomi inglesi che in italiano rimangono singolari, modi di dire che, una volta tradotti letteralmente, non ti escono più dalla testa. Spero, comunque, che riusciate a capire in modo più agevole questa correzione rimanendo, come sempre, a disposizione. David (link at the beginning of the document)