Copias de seguridade: tar e rsync





Nesta sección estudaranse tipos de copias de seguridade (**total**, **diferencial** ou **incremental**), métodos para realizar un plan de copias de seguridade (**Avó, pai e fillo** ou as **torres de hanoi**), utilidades para realizar copia de datos (**tar** ou **rsync**) e presentarase un script para implantar con **tar** un sistema de backup baseado na estratexia de **Avó, pai e fillo** (GFS, *Grandfather, father and son*).

LEMBRAR EN UBUNTU DESKTOP

Antes de comezar con esta sección é aconsellable que o usuario estea familiarizado con copiar a CDs, imaxes iso, arquivos empaquetados ou comprimidos, etc.

- Copias de seguridade: datos, música, ficheiros comprimidos e backups

do curso Curso de instalación de Ubuntu nun sistema dual.

Sumario

- 1 Introdución
- 2 Métodos de copia de seguridade
- 3 Estratexias de copias de seguridade
  - 3.1 Simple
  - 3.2 Pai ? Fillo (FS: Father - Son)
  - 3.3 Avó ? Pai ? Fillo (GFS: Grandfather - Father -Son)
  - 3.4 Torres de Hanói
    - 3.4.1 Torres de Hanói: 3 cintas

        - ◊ 3.4.2 Torres de Hanói: 4 cintas
        - ◊ 3.4.3 Torres de Hanói: 5 cintas
    - ♦ 3.5 Torres de Hanói: 5 cintas dispostas en formato mes
- 4 Tar
    - ♦ 4.1 Tipos de compresión
    - ♦ 4.2 Sintaxe do comando tar
    - ♦ 4.3 Backups incrementais e diferencias con tar
        - ◊ 4.3.1 Copias Incrementais
        - ◊ 4.3.2 Copias Diferenciais
- 5 Script usando tar e o método GFS
    - ♦ 5.1 Programar as copias de seguridade
- 6 Rsync
    - ♦ 6.1 Exemplo: sincronización local
    - ♦ 6.2 Exemplo: sincronización remota
- 7 Instantánea escenario 2.C - Backup

Introdución

- De todos é sabido a importancia que ten que se realicen copias de seguridade coa maior frecuencia posible para poder recuperarse ante posibles desastres ou perdas de información.

- Aínda que existen moitas ferramentas para realizar copias de seguridade, moitas delas facendo uso de bases de datos para levar rexistro das copias realizadas e do que contén cada copia (podemos citar como exemplo duplicity como ferramenta libre e gratuíta que permite almacenar as copias en rede e na nube cifrando o seu contido), aquí vanse usar dúas utilidades moi sinxelas, á par que potentes:
    - ♦ **tar**: para empaquetar arquivos e comprimilos.
    - ♦ **rsync**: para transmitir eficientemente arquivos con crecemento incremental.

- Ademais para cubrir á maior parte de usuarios, as copias realizaranse a disco e non a cintas, lugar este último recomendable no que gardar as copias destinadas a perdurar no tempo.

Métodos de copia de seguridade

Segundo os arquivos que se copien para o backup, distínguense:

- **Copia completa ou total:** cópianse todos os arquivos seleccionados para a copia.
- **Copia diferencial:** só se copian os arquivos que se modificaron dende a última copia total. Na última copia diferencial sempre se copian os arquivos que se copiaron nas anteriores copias diferenciais.
- **Copia incremental:** só se copian os arquivos que se modificaron con respecto á última copia, sexa esta total, diferencial ou incremental.

O ideal sería ter unha copia completa todos os días, pero iso consume moito espazo e tempo para crear o backup. En cambio para restaurar ante unha continxencia sería o método máis rápido, xa que só habería que restaurar o último día que se fixo a copia completa.

- Unha estratexia de copias de seguridade pode combinar os tres métodos anteriores: por exemplo:
    - ♦ **Total+diferenciais**: Luns total, de martes a venres diferenciais.
    - ♦ **Total+incrementais**: Luns total, de martes a venres incrementais.

- Vantaxes e inconvenientes das combinacións anteriores, supoñendo que o xoves houbo unha catástrofe.:
    - ♦ **Total+diferenciais**: cada diferencial consume cada vez máis tempo e espazo, pola contra para recuperar o sistema só habería que recuperar a copia total do luns e a diferencial do mércores.
    - ♦ **Total+incrementais**: cada incremental ten só os arquivos que se modificaron con respecto á anterior copia, sexa esta total ou incremetal, pola contra para recuperar o sistema levaría máis tempo, habería que recuperar a copia total do luns, a incremental do martes e finalmente a incremental do mércores.

Estratexias de copias de seguridade

A continuación explicaranse estratexias que combinan os métodos anteriores (non se ten en conta a fin de semana, aínda que os cambios para contemplala son moi intuitivos).

Simple

Unha copia total pola noite cada día hábil da semana. Cada copia dese día da semana machaca a copia dese mesmo día da semana anterior. A copia do luns desta semana machacará a copia do luns da semana anterior:

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia	1	2	3	4	5	-	-

Cada día habería que sacar da oficina a copia do día anterior e traer a dese día. Deste xeito tamén se prevén contra furtos e catástrofes como incendios, derrubamentos, inundacións, etc. que de acontecer e afectar ao sistema informático da empresa tamén afectarían ás copias que estivesen dentro da empresa ou oficina.

- **Os pros**: moi sinxela
- **Os contras**: o backup histórico é moi curto.

Pai ? Fillo (FS: Father - Son)

Unha copia incremental ou diferencial (mellor esta última) pola noite de luns a xoves e unha total aos venres. As copias de luns a xoves machácanse coas da semana seguinte e a do venres machácase cada 15 días.

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	1	2	3	4	5	-	-
copia sem-2	1	2	3	4	6	-	-

- Número de copias/cintas/ficheiros ao mes: 6
  - **Fillos:** diferenciais ou incrementais: copias 1, 2, 3 e 4
  - **Pai**: completas ou totais: copias 5 e 6
- **Os pros**: sinxela, tense un histórico de 8 días atrás.
- **Os contras**: o backup histórico só é 8 días.

Esta estratexia, poderíase mellorar cunha copia completa por mes.

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	1	2	3	4	5	-	-
copia sem-2	1	2	3	4	6-m	-	-

- Copia m: sería unha copia completa cada último venres de cada mes. Cantas copias mensuais?, depende do histórico que se desexe ter. Con tres cintas, por exemplo, para a copia mensual, (9 en total) só se tería o histórico de 3 meses e medio.

Avó ? Pai ? Fillo (GFS: Grandfather - Father -Son)

Unha copia incremental ou diferencial (mellor esta última) pola noite de luns a xoves, unha total por semana, aos venres e unha total mensual o último venres de cada mes.

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	1	2	3	4	5	-	-
copia sem-2	1	2	3	4	6	-	-
copia sem-3	1	2	3	4	7	-	-
copia sem-4	1	2	3	4	8-m	-	-

- Número de copias/cintas/ficheiros ao mes: 9
  - **Fillos:** diferenciais ou incrementais diarias: copias 1, 2, 3 e 4
  - **Pai**: completas ou totais semanais: copias 5, 6, 7 e 8 se hai quinto venres no mes.
  - **Avó**: completas ou totais mensual: copias m (último venres de cada mes).

- **Os pros**: Aumenta a redundancia, agora tense un mes de historia, e teranse tantos meses como copias m distintas se desexen realizar, o ideal 12 copias m.
- **Os contras**: ao haber máis copias o esquema comeza a complicarse.

- Hai outra variación desta estratexia, que consiste en facer a copia mensual o último día hábil de cada mes:

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	1	2	3	4	5	-	-
copia sem-2	1	2	3	4	6	-	-
copia sem-3	1	2	3	4	7	-	-
copia sem-4	1	2	3	4	8	-	-
copia sem-5	1	m	3	4	5	-	-

O ideal, en calquera dos casos, é ter 12 copias m, e estas gardalas fóra da oficina, polo que xa se comentou anteriormente.

Torres de Hanói

Baséase no xogo matemático das torres de Hanói

- As Torres de Hanoi é unha estratexia de backup cun método de rotación máis complexo. Esencialmente é un método recursivo. É unha forma 'intelixente' de arquivar un número efectivo de backups totais ou completos, así como de dispor de históricos, pero é máis complexo de entender.

- Un conxunto de *n* cintas ou discos permite realizar 2^{n-1} días de backup antes de comezar un novo ciclo de backups. Así con 3 cintas cóbrense 4 días de backup, con 4 cintas 8 días de backup, con 5 cintas 16 días, con 6 32 días, con 7 cintas 64 días, ...

Basicamente, cada cinta/disco está asociada cun disco do puzzle, e cada movemento do disco a un pau diferente corresponde a un backup para a cinta/disco. Así, a primeira cinta úsase cada 2 días todos os días (1, 3, 5, 7, 9 ,...), a segunda cinta úsase cada 4 días (2, 6, 10, ...), a terceira cinta cada oitavo día (4, 12, 20, ...).

As táboas seguintes amosan que cintas son usadas e en que días dentro de cada ciclo. Fixarse como a primeira cinta é sobreescrita cada 2 días. O ideal é comezar coa última cinta do ciclo, a marcada en vermello.

Torres de Hanói: 3 cintas

	Día do ciclo							
	01	02	03	04	05	06	07	08
Cintas	A		A		A		A	
		B				B		
				C				C

Torres de Hanói: 4 cintas

	Día do ciclo															
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Cintas	A		A		A		A		A		A		A		A	
		B				B				B				B		
				C								C				
								D								D

Torres de Hanói: 5 cintas

	Día do ciclo																													
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Día	A		A		A		A		A		A		A		A		A		A		A		A		A		A		A		A	
		B				B				B				B				B				B				B				B		
				C								C								C								C				
								D																D							
																E																E

Torres de Hanói: 5 cintas dispostas en formato mes

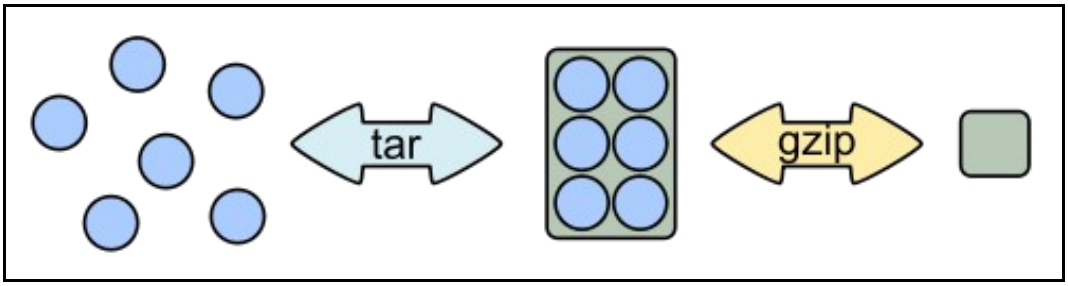
	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	A	B	A	C	A	-	-
copia sem-2	B	A	D	A	B	-	-
copia sem-3	A	C	A	B	A	-	-
copia sem-4	E	A	B	A	C	-	-
copia sem-5	A	B	A	D	A	-	-

- **Os pros**: Con moi poucas cintas pódese conseguir un histórico moi considerable.
- **Os contras**: a parte de que a cinta A acabará esgotándose máis pronto que as demais, fai falla dicir os contras?, pois iso, hai que ter todo moi ben etiquetadiño, un calendario moi ben revisado, e ser moi sistemáticos e pulcros na operativa desta estratexia.

Tar

Tar (Tape Archiver) é unha utilidade que basicamente serve para empaquetar varios arquivos nun só arquivo contedor, que opcionalmente pódese comprimir.

A seguinte imaxe amosa unha síntese do proceso de empaquetado (tar) e compresión (neste caso gzip).



A utilidade **tar** non debe ser usado cando hai arquivos abertos ou en uso por programas.

Tipos de compresión

A parte de empaquetar varios arquivos nun só arquivo contedor, este arquivo resultante pode ser comprimido para aforrar espazo: para iso existen varios algoritmos libres, canda un coas súas vantaxes e inconvenientes: cando comprime pouco é moi rápido, pola contra cando comprime moito é moi lento e consume moita cpu e memoria.

Nesta páxina da wikipedia explícase o funcionamento de varios compresores libres.

Nesta outra páxina faise unha comparativa entre factor de compresión e tempo consumido dos algoritmos: gzip (.gz), bzip (.bz2) e LZMA (.lzma). Este último é o que usa o programa 7-Zip.

Na páxina do comando tar indícanse que algoritmos de compresión son soportados por esta utilidade.

A diferenza básica entre os tres formatos enriba mencionados é a seguinte:

- **.gz:** usa gzip, menor compresión alta velocidade, ficheiro resultante máis grande.
- **.bz2:** usa bzip2, boa compresión, velocidade moderada, ficheiro resultante tamaño moderado.
- **.lzma:** usa lzma, alta compresión, baixa velocidade, o ficheiro resultante é o máis pequeno.

Sintaxe do comando tar

A sintaxe do comando é como segue:

```
tar <opcións> <arquivoSaída> <arquivo1> <arquivo2> ... <arquivoN>
```

```
opcións máis usadas:
#-c: para crear un arquivo contedor
#-f: para especificar o nome do arquivo contedor
#-v: amosa información do que está a facer
#-x: descomprime/extrae información
#-t: lista o contido dun arquivo
#-z: comprime/descomprime con gzip
#-j: comprime/descomprime con bzip2
#?lzma: comprime/descomprime con lzma
#-a: selecciona o algoritmo de compresión pola extensión do arquivo.
```

Exemplos:

Para realizar os seguintes exemplos iniciouse sesión en **uclient01** coa usuario **sol** e creáronse dous directorios e ficheiros de exemplo dentro deles, como se amosa no primeiro exemplo seguinte.

```
#Crear un empaquetado amosando o que fai
sol@uclient01:~$ tar cvf empaquetado1.tar dir1 dir2
dir1/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/
dir2/a1.txt
dir2/a2.txt


#Listar o contido dun empaquetado
sol@uclient01:~$ tar tf empaquetado1.tar
dir1/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/
dir2/a1.txt
dir2/a2.txt


# Crear un tar e logo comprimilo usando gzip
sol@uclient01:~$ tar czf empaquetado2.tar.gz dir1 dir2


# Crear un tar e comprimilo autodetectando o formato pola extensión
sol@uclient01:~$ tar cfa empaquetado3.tar.lzma dir1 dir2

#Canto ocupa cada paquete?
sol@uclient01:~$ ls empaquetado* -lh
-rw-r--r-- 1 sol g-profes 20K Mai  8 22:35 empaquetado1.tar
-rw-r--r-- 1 sol g-profes 2,5K Mai  8 22:36 empaquetado2.tar.gz
-rw-r--r-- 1 sol g-profes 2,1K Mai  8 22:36 empaquetado3.tar.lzma

sol@uclient01:~$ rm dir1 -r
sol@uclient01:~$ rm dir2 -r

#Recuperar dir1 e dir2. As dúas sentencias seguintes son equivalentes.
sol@uclient01:~$ tar xfa empaquetado3.tar.lzma
sol@uclient01:~$ tar xf empaquetado3.tar.lzma

sol@uclient01:~$ rm dir1 -r

#Recuperar só dir1
sol@uclient01:~$ tar xf empaquetado3.tar.lzma dir1
```

Backups incrementais e diferencias con tar

Con tar tamén se poden realizar copias incrementais, e cun chisco de operacións máis copias diferenciais.

Para iso usarase o parámetro: **--listed-incremental=ficheiro.snar** ou **-g**. O ficheiro *.snar é un arquivo que contén metadatos sobre os arquivos que están no empaquetado. Este ficheiro chámase **snapshot**, o seu obxectivo é axudar a determinar que ficheiros cambiaron, se engadiron ou borraron

dende o último backup (sexa este total ou incremental).

O seu formato é:

```
tar cf <arquivoSaída> -g snapshot_file.snar <arquivo1> <arquivo2> ...
```

- **Consideracións sobre nfs:** Cando se crea o *snapshot_file* tar almacena información sobre o tempo, selos temporais e o identificador do dispositivo no que están os directorios que se copian. En nfs, este identificador cambia cada vez que se monta un recurso nfs. Por tanto tar sempre fará copias completas salvo que se especifique a opción **no-check-device** antes dos directorios/arquivos a copiar que están sobre nfs.

```
tar cf <arquivoSaída> -g snapshot_file.snar --no-check-device <arquivo1> <arquivo2> ...
```

- Esta opción non é necesaria cando a copia se fai de sistemas de almacenamento fixos, que teñen sempre o mesmo UUID.

Copias Incrementais

- Para amosar o seu funcionamento vaise entrar en **uclient01** coa usuario **sol**, faranse backups de dous directorios da súa carpeta persoal (co cal vai ser un tar sobre nfs).

- Caso de estudio, quérese ter tres arquivos de backup:
  - **backup_total**: ten unha copia total
  - **backup_inc_1**: contén os arquivos que se modificaron dende a copia total
  - **backup_inc_2**: contén os arquivos que se modificaron dende a anterior incremental.

Logo volverase a comezar o ciclo, machacando os arquivos de backup.

- **Primeiro ciclo de copias:**

```
#Primeira copia completa, lembrar que se está sobre nfs e hai que especificar --no-check-device

sol@uclient01:~$ tar cvfa backup_total.tar.bz2 -g backup.snar --no-check-device dir1 dir2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt


#Modifícase un ficheiro e créase un subdirectorio.
sol@uclient01:~$ nano dir1/f1.txt
sol@uclient01:~$ mkdir dir1/subdir

# Faise a primeira copia incremental.
# O arquivo backup.snar é lido para ver que se modificou dende a última copia.
# Ó mesmo tempo ese arquivo é modificado, contendo información sobre a nova copia.
# A copia incremental 1 só contén o que variou con respecto á anterior copia completa.
sol@uclient01:~$ tar cvfa backup_inc_1.tar.bz2 -g backup.snar --no-check-device dir1 dir2
tar: dir1/subdir: O directorio é novo
dir1/
dir2/
dir1/subdir/
dir1/f1.txt


#Modifícase un segundo arquivo.
sol@uclient01:~$ nano dir2/a1.txt

#A copia incremental 2 só ten o que se modificou dende que se fixo a incremental 1.
#Volve a modificarse o arquivo de metadatos backup.snar cos datos da segunda copia incremental.
sol@uclient01:~$ tar cvfa backup_inc_2.tar.bz2 -g backup.snar --no-check-device dir1 dir2
dir1/
dir2/
dir1/subdir/
dir2/a1.txt
```

```
#Arquivos creados. Fixarse nos tamaños dos arquivos.
sol@uclient01:~$ ls back* -lh
-rw-r--r-- 1 sol g-profes  316 2010-03-22 10:39 backup_inc_1.tar.bz2
-rw-r--r-- 1 sol g-profes  305 2010-03-22 10:40 backup_inc_2.tar.bz2
-rw-r--r-- 1 sol g-profes  211 2010-03-22 10:40 backup.snar
-rw-r--r-- 1 sol g-profes 2,6K 2010-03-22 10:38 backup_total.tar.bz2


#O contido de cada un dos tres arquivos comprimidos.
sol@uclient01:~$ tar tf backup_total.tar.bz2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

sol@uclient01:~$ tar tf backup_inc_1.tar.bz2
dir1/
dir2/
dir1/subdir/
dir1/f1.txt

sol@uclient01:~$ tar tf backup_inc_2.tar.bz2
dir1/
dir2/
dir1/subdir/
dir2/a1.txt
```

- **Segundo e seguintes ciclos de copias**.

Como se viu no exemplo anterior o snapshot_file *backup.snar* vai cambiando cada vez que se realiza unha copia, co cal se se volve a usar para facer a copia completa que corresponde ao segundo ciclo de copias (4º día) vai facer unha incremental no canto dunha total.

- Solución:
  - ate a versión 1.23 de tar: borrar o arquivo backup.snar, que se non existe vólvese a crear, co cal cada vez que comece un novo ciclo de copias débese borrar. A versión que vén con Ubuntu 14.04 é a 1.27 e con Debian a 1.26
  - Versión 1.23 de tar: hai un novo parámetro: **--level** que forza o nivel de copia. Level=0 cópiase todo, level=1 primeira copia incremental, ...

Co cal a solución sería:

```
tar cf backup_total.tar.bz2 -g backup.snar --level=0 --no-check-device dir1 dir2
```

- Imos alá supoñendo que se dispón dunha versión de tar anterior á 1.23.

```
#borrar o arquivo de metadatos cada vez que se comeza un novo ciclo de copias
sol@uclient01:~$ rm backup.snar

#Copia total do segundo ciclo de copias
sol@uclient01:~$ tar cvfa backup_total.tar.bz2 -g backup.snar --no-check-device dir1 dir2
tar: dir1: O directorio é novo
tar: dir1/subdir: O directorio é novo
tar: dir2: O directorio é novo
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

#Arquivos implicados na copia
sol@uclient01:~$ ls backup* -lh
-rw-r--r-- 1 sol g-profes  316 2010-03-22 10:39 backup_inc_1.tar.bz2
```

```
-rw-r--r-- 1 sol g-profes  305 2010-03-22 10:40 backup_inc_2.tar.bz2
-rw-r--r-- 1 sol g-profes  211 2010-03-22 11:22 backup.snar
-rw-r--r-- 1 sol g-profes 2,7K 2010-03-22 11:22 backup_total.tar.bz2

#Contido da copia completa actual.
sol@uclient01:~$ tar tf backup_total.tar.bz2
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

# E agora tocaria xa facer as 2 incrementais sen máis.
```

- **Restaurar ante catástrofe**.

O proceso de restauración dun ficheiro tar é igual que o normal, non fai falla o arquivo de metadatos.

Imaxinar que o cuarto día, antes de facer a copia total do segundo ciclo de copias, hai unha continxencia nos datos e hai que restaurar: Habería que extraer os arquivos da total do primeiro día, a incremental 1 e a incremental 2, para deixar o sistema o máis parecido posible ó momento antes da catástrofe.

Copias Diferenciais

Como se viu no exemplo anterior, o arquivo de metadatos backup.snar vai cambiando cada vez que se realiza unha nova copia de datos. Para unha copia diferencial interesa saber que foi o que cambiou dende a última copia total, bueno pois antes de facer cada incremental copiase o backup.snar da copia total, para mirar as diferenzas en cada copia con respecto á total.

- Caso de estudio, quérese ter tres arquivos de backup:
  - **backup_total**: ten unha copia total
  - **backup_dif_1**: contén os arquivos que se modificaron dende a copia total.
  - **backup_dif_2**: contén os arquivos que se modificaron dende a copia total, inclúe os que están en backup_dif_1.

Logo volverase a comezar o ciclo, machacando os arquivos de backup.

- **Primeiro ciclo de copias:**

```
#Primeira copia total
sol@uclient01:~$ tar cvfa backup_total.tar.bz2 -g backup.snar --no-check-device dir1 dir2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

#Modificación ficheiro f1.txt, crear dir2/a3.txt e un subdirectorio
sol@uclient01:~$ nano dir1/f1.txt
sol@uclient01:~$ nano dir2/a3.txt
sol@uclient01:~$ mkdir dir1/subdir

#Copiar o arquivo de metadatos para primeira copia diferencial e usalo no tar.
sol@uclient01:~$ cp backup.snar backup_dif_1.snar
sol@uclient01:~$ tar cvfa backup_dif_1.tar.bz2 -g backup_dif_1.snar --no-check-device dir1 dir2
tar: dir1/subdir: O directorio é novo
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir2/a3.txt
```

```
#Modificar o ficheiro f2.txt
sol@uclient01:~$ nano dir1/f2.txt

#Copiar o arquivo de metadatos para a segunda copia diferencial e usalo no tar.
#Observar como se copian tamén os arquivos/directorios que tamén se crearon na diferecial 1.
sol@uclient01:~$ cp backup.snar backup_dif_2.snar
sol@uclient01:~$ tar cvfa backup_dif_2.tar.bz2 -g backup_dif_2.snar --no-check-device dir1 dir2
tar: dir1/subdir: O directorio é novo
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir1/f2.txt
dir2/a3.txt


#Os arquivos implicados, fixarse como a cada diferencial o tamaño vai crecendo.
sol@uclient01:~$ ls back* -lh
-rw-r--r-- 1 sol g-profes  219 2010-03-22 11:54 backup_dif_1.snar
-rw-r--r-- 1 sol g-profes  388 2010-03-22 11:54 backup_dif_1.tar.bz2
-rw-r--r-- 1 sol g-profes  219 2010-03-22 11:55 backup_dif_2.snar
-rw-r--r-- 1 sol g-profes  609 2010-03-22 11:55 backup_dif_2.tar.bz2
-rw-r--r-- 1 sol g-profes  156 2010-03-22 11:52 backup.snar
-rw-r--r-- 1 sol g-profes 2,7K 2010-03-22 11:52 backup_total.tar.bz2

#O contido de cada arquivo da copia
sol@uclient01:~$ tar tf backup_total.tar.bz2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

sol@uclient01:~$ tar tf backup_dif_1.tar.bz2
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir2/a3.txt

sol@uclient01:~$ tar tf backup_dif_2.tar.bz2
dir1/
dir1/subdir/
dir2/
dir1/f1.txt
dir1/f2.txt
dir2/a3.txt
sol@uclient01:~$
```

- **Segundo ciclo de copias e seguintes**

Aplícase o indicado na copia incremental.

- **Restaurar ante catástrofe**.

O proceso de restauración dun ficheiro tar é igual que o normal, non fai falla o arquivo de metadatos.

Imaxinar que o cuarto día, antes de facer a copia total do segundo ciclo de copias, hai unha continxencia nos datos e hai que restaurar: Habería que extraer os arquivos da total do primeiro día, a diferencial 2, para deixar o sistema o máis parecido posible ó momento antes da catástrofe. Non fai falla usar a diferencial 1.

Script usando tar e o método GFS

O seguinte script foi creado para realizar copias de seguridade usando a primeira versión da estratexia Avó, Pai e Fillo:

	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1	1	2	3	4	5	-	-
copia sem-2	1	2	3	4	6	-	-
copia sem-3	1	2	3	4	7	-	-
copia sem-4	1	2	3	4	8-m	-	-

Se hai días hábiles despois do 4º venres pero sen chegar ate ó 5º venres, eses días irán na copia mensual do próximo mes.

- Este script crea un directorio para o ano en curso e dentro del:
  - 1 A primeira copia total
  - 3 ou 4 copias semanais do último mes
  - 1 copia mensual por cada mes.

Este script pode adaptarse a outras situacións. O propio script está autodocumentado para poder seguilo mellor.

```
#!/bin/sh



#..../backup.sh


#Variables configurables polo usuario ------------------------------------------------

ORDENADOR=`hostname`# Nome do ordenador que executa o script
# Obrigatorio poñer algo



BACKUP_DIR="backup"# Onde se vai realizar o backup. Debe existir
# Obrigatorio


DIRECTORIOS="/dir1/sub /dir2"# Directorios a copiar
# Obrigatorio


N_C_D=""# Se BACKUP_DIR está sobre almacenamento:
# nfs: --no-check-device, para que non teña
#     en conta o UUID no arquivo de metadatos.
#
# sistema fixo: nada, para que colla o UUID


SNAR=snar         # subdirectorio de BACKUP_DIR no que gardar
# os arquivos de metadatos.
# Opcional, posibles valores
# nada ou . -> backups e metadatos no mesmo
#              directorio: BACKUP_DIR
# directorio -> subdirectorio de BACKUP_DIR



COMPRESOR=.lzma        # Tar ten o parámetro -a para que en función
# da extensión use o compresor que corresponda
# posibles valores para COMPRESOR (opcional):
# nada -> non comprime, só empaqueta
# .gz  -> usa gzip, menor compresión
#                   alta velocidade
# .bz2 -> usa bzip2, boa compresión
#                   velocidade moderada
```

```
# .lzma -> usa lzma, alta compresión
#                   baixa velocidade
# consultar http://www.gnu.org/software/tar
# para máis opcións




N_DIA_SEMANA_COPIA_TOTAL=5# En que número de día da semana se desexa
# realizar a copia Total. 5 (ven), 6 (sáb),etc.
# Nos días de semana superiores a este non se
# realizaran copias, salvo que non exista
# ningunha copia e sexa a primeira vez que se
# executa o script, entón si que fará a
# primeira copia.
# Máis vale previr que lamentar!!!
# Obrigatorio



# Variables do script -------------------------------------------------------------
N_MES=`date +%m`# Número Mes actual:    01 .. 12

MES_ABREV=`date +%b`# Nome Mes abreviado:   Xan .. Dec

N_DIA_MES=`date +%d`# Número do día do mes: 1 .. 31

N_DIA_SEMANA=`date +%u`# Número de día da semana:      1 .. 7

DIA_SEMANA_ABREV=`date +%a`# Nome día semana abreviado:    Lun .. Dom

COPIA_MENSUAL='N'# Flag para controlar se a copia é mensual.


ANO=`date +%Y`# Obter os dous últimos díxitos do ano.

BACKUP_DIR=$BACKUP_DIR"/"$ANO# Redefinir o directorio de backup.


F_METADATOS_ULTIMA_TOTAL=$BACKUP_DIR/$SNAR/$ORDENADOR-00-ULTIMA_TOTAL.snar
# Ten a ruta ó ficheiro de metadatos da última
# copia Total.

REXISTRO_COPIAS=$BACKUP_DIR/$SNAR/rexistro.txt
# Neste ficheiro vaise levar un log das copias
# que se realizan. Cando comezan, tipo, cando
# rematan.



#---- SCRIPT ----

if ! test -d "$BACKUP_DIR/$SNAR";then
        mkdir -p "$BACKUP_DIR/$SNAR"
fi


# Comprobar se hoxe é o día de facer unha copia Total

if [ $N_DIA_SEMANA = $N_DIA_SEMANA_COPIA_TOTAL ]; then

# En que semana do mes se vai realizar a copia semanal Total ou mensual
case $N_DIA_MES in

0[1-7])       SEMANA=Semana_1;;

08|09|1[0-4]) SEMANA=Semana_2;;

1[5-9]|20|21) SEMANA=Semana_3;;

      # Se a semana é a 4, hai que saber se dentro
      # de 7 días se está no mesmo mes (habería unha
```

```
                              # semana 5) ou non (habería que facer a copia
      # mensual)

2[2-8])       SEMANA=Semana_4;
      if [ $N_MES != `date -d "7 days" +%m` ]
      then
         COPIA_MENSUAL='S'
      fi;;

29|30|31)     SEMANA=Semana_5;
      COPIA_MENSUAL='S';;
esac


# Se a copia é mensual hai que facer unha Total do mes.

if [ $COPIA_MENSUAL = 'S' ];then

echo `date` "Avó     :$ORDENADOR - INICIO: Total mensual $MES_ABREV">>$REXISTRO_COPIAS

F_BACKUP=$BACKUP_DIR/$ORDENADOR-01-Total-Mes-$N_MES-$MES_ABREV.tar$COMPRESOR
        F_METADATOS=$BACKUP_DIR/$SNAR/$ORDENADOR-01-Total-Mes-$N_MES-$MES_ABREV.snar
else
echo `date` "Pai     :$ORDENADOR - INICIO: Total $SEMANA">>$REXISTRO_COPIAS

F_BACKUP=$BACKUP_DIR/$ORDENADOR-02-Total-$SEMANA.tar$COMPRESOR
F_METADATOS=$BACKUP_DIR/$SNAR/$ORDENADOR-02-Total-$SEMANA.snar
fi

# Se existe o arquivo de metadatos hai que borralo. para que faga unha
# copia completa.

if test -f "$F_METADATOS";then
                rm $F_METADATOS
fi

# Realizase a copia total, en función das variables enriba especificadas
# Cópiase o arquivo de metadatos, para poder realizar logo as copias
# diarias diferenciais.

        $BACKUP -g $F_METADATOS $N_C_D $DIRECTORIOS
   cp $F_METADATOS $F_METADATOS_ULTIMA_TOTAL

echo `date` "Avó-Pai :$ORDENADOR - FIN   : Total">>$REXISTRO_COPIAS
echo >>$REXISTRO_COPIAS


# Cada mes métese unha separación no arquivo de rexistro de copias
if [ $COPIA_MENSUAL = 'S' ];then
echo >>$REXISTRO_COPIAS
echo >>$REXISTRO_COPIAS
echo ---------- MES de  `date -d "7 days" +%B` ------------->>$REXISTRO_COPIAS
fi



# PRIMEIRA COPIA TOTAL

# Se non existe unha copia Total, entón é a primeira vez que se executa este
# script e ademais, non é o día que tocaría facer unha copia semanal ou mensual.
# Neste caso vaise crear unha copia Total, aínda que no día en que se realice
# non se debera facer copias ou tocara unha diferencial.
# Máis vale previr que lamentar!!!


elif ! test -f "$F_METADATOS_ULTIMA_TOTAL";then

echo `date` "Primeira:$ORDENADOR - INICIO: Total. $DIA_SEMANA_ABREV">>$REXISTRO_COPIAS

F_BACKUP=$BACKUP_DIR/$ORDENADOR-00-Primeira-$DIA_SEMANA_ABREV.tar$COMPRESOR
F_METADATOS=$F_METADATOS_ULTIMA_TOTAL

        $BACKUP -g $F_METADATOS $N_C_D $DIRECTORIOS
```

```
echo `date` "Primeira:$ORDENADOR - FIN   : Total. $DIAS_SEMANA_ABREV">>$REXISTRO_COPIAS
echo >>$REXISTRO_COPIAS



# Se é un día de semana anterior ó que se realiza a copia semanal total.

elif [ $N_DIA_SEMANA -lt $N_DIA_SEMANA_COPIA_TOTAL ];then

echo `date` "Fillo   :$ORDENADOR - INICIO: Diferencial. Día $DIA_SEMANA_ABREV">>$REXISTRO_COPIAS

F_BACKUP=$BACKUP_DIR/$ORDENADOR-03-Dif$N_DIA_SEMANA-$DIA_SEMANA_ABREV.tar$COMPRESOR
F_METADATOS=$BACKUP_DIR/$SNAR/$ORDENADOR-03-Dif$N_DIA_SEMANA-$DIA_SEMANA_ABREV.snar



# Copiar o arquivo de metadatos da última total, para o día de hoxe.

   $F_METADATOS_ULTIMA_TOTAL $F_METADATOS
        $F_BACKUP -g $F_METADATOS $N_C_D $DIRECTORIOS

echo `date` "Fillo   :$ORDENADOR - FIN   : Diferencial. Día $DIA_SEMANA_ABREV">>$REXISTRO_COPIAS
echo >>$REXISTRO_COPIAS


else

echo `date` "Non     :$ORDENADOR - Ós $DIA_SEMANA_ABREV non se realizan backups">>$REXISTRO_COPIAS
echo >>$REXISTRO_COPIAS

fi
```

Programar as copias de seguridade

O ideal é ter o script anterior en */usr/local/sbin/backup.sh* e chamalo dende unha tarefa programada.

crontab de root: (crontab -e)

```
LANG=gl_ES.UTF-8

# m  h  dom  mon dow command
  55 23  *   *    *  /usr/local/sbin/backup.sh
```

- Para probar o script fixéronse cambios no mesmo nos seguintes campos:

```
BACKUP_DIR="backup"             # Onde se vai realizar o backup. Debe existir
                                # Obrigatorio

DIRECTORIOS="dir1 dir2"         # Directorios a copiar que están no cartafol de root
                                # Obrigatorio
```

Creouse un crontab para root e executouse como este mesmo usuario.

A imaxe amosa o calendario do 2010.

```
root@dserver00:~# cal 2010
                                2010
      Xaneiro                   Febreiro                  Marzo
Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá
                1  2      1  2  3  4  5  6      1  2  3  4  5  6
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   7  8  9 10 11 12 13
10 11 12 13 14 15 16  14 15 16 17 18 19 20  14 15 16 17 18 19 20
17 18 19 20 21 22 23  21 22 23 24 25 26 27  21 22 23 24 25 26 27
24 25 26 27 28 29 30  28                    28 29 30 31
31

       Abril                     Maio                      Xuño
Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá
             1  2  3                     1            1  2  3  4  5
 4  5  6  7  8  9 10   2  3  4  5  6  7  8   6  7  8  9 10 11 12
11 12 13 14 15 16 17   9 10 11 12 13 14 15  13 14 15 16 17 18 19
18 19 20 21 22 23 24  16 17 18 19 20 21 22  20 21 22 23 24 25 26
25 26 27 28 29 30     23 24 25 26 27 28 29  27 28 29 30
                      30 31

       Xullo                     Agosto                    Setembro
Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá
             1  2  3   1  2  3  4  5  6  7            1  2  3  4
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   5  6  7  8  9 10 11
11 12 13 14 15 16 17  15 16 17 18 19 20 21  12 13 14 15 16 17 18
18 19 20 21 22 23 24  22 23 24 25 26 27 28  19 20 21 22 23 24 25
25 26 27 28 29 30 31  29 30 31              26 27 28 29 30

      Outubro                   Novembro                  Decembro
Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá  Do Lu Ma Mé Xo Ve Sá
                1  2      1  2  3  4  5  6            1  2  3  4
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   5  6  7  8  9 10 11
10 11 12 13 14 15 16  14 15 16 17 18 19 20  12 13 14 15 16 17 18
17 18 19 20 21 22 23  21 22 23 24 25 26 27  19 20 21 22 23 24 25
24 25 26 27 28 29 30  28 29 30              26 27 28 29 30 31
31
```

O script comezouse a executar o 13 de marzo (un sábado, que segundo a configuración do script non se realizan backups, pero como era a primeira execución...), e o último día que se executou foi o luns 29 de marzo. As modificacións mensuais dos días 29, 30 e 31 de marzo serán recollidas na copia mensual do mes de abril (30 de abril).

- Co cal tense, do 13 ao 29 de marzo:
  - 1 copia semanal: a do 19 de marzo
  - 4 copias diferenciais: a do luns 29, martes 23, mércores 24, xoves 26.
  - 1 copia mensual: a do 26 de marzo.

```
#O comando tree actúa igual que en MS-DOS, hai que instalalo.
root@dserver00:~$ tree backup/2010 -hND
backup/2010
`-- [4.0K Mar 26 23:55]  2010
    |-- [2.6K Mar 13 23:55]  dserver00-00-Primeira-Sáb.tar.bz2
    |-- [4.7K Mar 26 23:55]  dserver00-01-Total-Mes-03-Mar.tar.bz2
    |-- [3.9K Mar 19 23:55]  dserver00-02-Total-Semana_3.tar.bz2
    |-- [3.3K Mar 29 23:55]  dserver00-03-Dif1-Lun.tar.bz2
    |-- [4.0K Mar 23 23:55]  dserver00-03-Dif2-Mar.tar.bz2
    |-- [4.1K Mar 24 23:55]  dserver00-03-Dif3-Mér.tar.bz2
    |-- [4.0K Mar 25 23:55]  dserver00-03-Dif4-Xov.tar.bz2
    `-- [4.0K Mar 26 23:55]  snar
        |-- [ 766 Mar 26 23:55]  dserver00-00-ULTIMA_TOTAL.snar
        |-- [ 766 Mar 26 23:55]  dserver00-01-Total-Mes-03-Mar.snar
        |-- [ 395 Mar 19 23:55]  dserver00-02-Total-Semana_3.snar
        |-- [ 872 Mar 29 23:55]  dserver00-03-Dif1-Lun.snar
        |-- [ 607 Mar 23 23:55]  dserver00-03-Dif2-Mar.snar
        |-- [ 660 Mar 24 23:55]  dserver00-03-Dif3-Mér.snar
        |-- [ 713 Mar 25 23:55]  dserver00-03-Dif4-Xov.snar
        `-- [2.0K Mar 29 23:55]  rexistro.txt

#Que amosa o ficheiro de rexistro de copias?, que se vai creando a medida que se fan as copias.
root@dserver00:~$ cat backup/2010/snar/rexistro.txt
Sáb Mar 13 23:55:00 CET 2010 Primeira:dserver00 - INICIO: Total. Sáb
Sáb Mar 13 23:55:00 CET 2010 Primeira:dserver00 - FIN   : Total.

Dom Mar 14 23:55:00 CET 2010 Non     :dserver00 - Ós Dom non se realizan backups

Lun Mar 15 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Lun
Lun Mar 15 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Lun

Mar Mar 16 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Mar
Mar Mar 16 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Mar

Mér Mar 17 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Mér
Mér Mar 17 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Mér

Xov Mar 18 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Xov
Xov Mar 18 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Xov

Ven Mar 19 23:55:00 CET 2010 Pai     :dserver00 - INICIO: Total Semana_3
Ven Mar 19 23:55:00 CET 2010 Avó-Pai :dserver00 - FIN   : Total

Sáb Mar 20 23:55:00 CET 2010 Non     :dserver00 - Ós Sáb non se realizan backups

Dom Mar 21 23:55:00 CET 2010 Non     :dserver00 - Ós Dom non se realizan backups

Lun Mar 22 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Lun
Lun Mar 22 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Lun

Mar Mar 23 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Mar
Mar Mar 23 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Mar

Mér Mar 24 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Mér
Mér Mar 24 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Mér

Xov Mar 25 23:55:00 CET 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Xov
Xov Mar 25 23:55:00 CET 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Xov

Ven Mar 26 23:55:00 CET 2010 Avó     :dserver00 - INICIO: Total mensual Mar
```

```
Ven Mar 26 23:55:00 CET 2010 Avó-Pai :dserver00 - FIN   : Total


---------- MES de Abril -------------
Sáb Mar 27 23:55:00 CET 2010 Non     :dserver00 - Ós Sáb non se realizan backups

Lun Mar 29 23:55:00 CEST 2010 Fillo   :dserver00 - INICIO: Diferencial. Día Lun
Lun Mar 29 23:55:00 CEST 2010 Fillo   :dserver00 - FIN   : Diferencial. Día Lun
```

Rsync

rsync é unha utilidade para manter sincronizados dous directorios ou arquivos. Mediante a técnica de Codificación Delta permite realizar sincronizacións simplemente enviando ao destino a diferenza entre os arquivos a sincronizar.

Se orixe e destino da sincronización están na mesma máquina, non usa a codificación delta, pois esta consume recursos, simplemente copia todo o que haxa en orixe. Se orixe e destino están e máquina distintas si que se usa esa técnica pois o que consume de recursos compensa coa redución de tempo que se obtén ó enviar pola rede un arquivo moito máis compacto e pequeno que o orixinal.

É moi recomendable usar rsync para ter sincronizadas as copias de seguridade feitas con tar, entre o servidor local e un remoto. E tamén se pode usar cron para programar esta tarefa.

- Formato do comando:

```
copias locais: rsync opcións orixe destino
copias remotas por ssh: rsync opcións -e ssh orixe destino
Nas copias remotas ou orixe ou destino= usuario@equipo:/directorio

# Opcións máis comúns:
# -a: mantén, propietarios, datas, permisos, etc dos arquivos/directorios orixe.
# -v: amosa o que fai
# --delete: borra en destino o que se borrara en orixe.
# -z: comprime os ficheiros no momento da transmisión do orixe ao destino. Útil en transmisións entre distintos ordenadores.
```

Exemplo: sincronización local

Trátase de sincronizar unha carpeta con outra, as dúas no mesmo equipo. Fixarse como desactiva a transmisión delta. Creamos antes unha carpeta, carpeta1 e con ficheiros dentro.

```
#A carpeta2 non existe e será creada por rsync
root@dserver00:~$ rsync -azvv carpeta1 carpeta2
sending incremental file list
created directory carpeta2
delta-transmission disabled for local transfer or --whole-file
carpeta1/
carpeta1/f4
carpeta1/f5
carpeta1/f6
total: matches=0  hash_hits=0  false_alarms=0 data=15264

sent 5587 bytes  received 73 bytes  11320.00 bytes/sec
total size is 15264  speedup is 2.70


#Modificar un ficheiro e borrar outro.
root@dserver00:~$ nano carpeta1/f4
root@dserver00:~$ rm carpeta1/f6

#Volver a sincronizar, nesta ocasión coa opción de ?delete.
root@dserver00:~$ rsync -azvv --delete carpeta1 carpeta2
sending incremental file list
```

```
delta-transmission disabled for local transfer or --whole-file
carpeta1/
deleting carpeta1/f6
carpeta1/f4
carpeta1/f5 is uptodate
total: matches=0  hash_hits=0  false_alarms=0 data=5098

sent 1940 bytes  received 38 bytes  3956.00 bytes/sec
total size is 10186  speedup is 5.15
```

Exemplo: sincronización remota

Usarase ssh, e para probar sen ter que usar outro equipo, o ssh realizarase contra o mesmo equipo. Fixarse como usa transmisión delta e como se toma o seu tempo antes de enviar.

```
#Sincronizar unha carpeta local cunha dun servidor. Carpeta3 non existe e será creada.
root@dserver00:~$ rsync --delete -azvv -e ssh carpeta1 root@localhost:/root/carpeta3
opening connection using: ssh -l root localhost rsync --server -vvlogDtprze.iLs --delete . /root/carpeta3
root@localhost\'s password:
sending incremental file list
created directory /root/carpeta3
delta-transmission enabled
carpeta1/
carpeta1/f4
carpeta1/f5
carpeta1/f6
total: matches=0  hash_hits=0  false_alarms=0 data=15274

sent 5602 bytes  received 73 bytes  1031.82 bytes/sec
total size is 15274  speedup is 2.69


# Editar un ficheiro e borrar outro.
root@dserver00:~$ nano carpeta1/f4
root@dserver00:~$ rm carpeta1/f6

# Volver a sincronizar.
root@dserver00:~$ rsync --delete -azvv -e ssh carpeta1 root@localhost:/root/carpeta3
opening connection using: ssh -l root localhost rsync --server -vvlogDtprze.iLs --delete . /root/carpeta3
root@localhost\'s password:
sending incremental file list
delta-transmission enabled
carpeta1/
deleting carpeta1/f6
carpeta1/f4
carpeta1/f5 is uptodate
total: matches=8  hash_hits=8  false_alarms=0 data=7

sent 160 bytes  received 86 bytes  54.67 bytes/sec
total size is 10193  speedup is 41.43
```

Instantánea escenario 2.C - Backup

- Ao igual que nos escenarios anteriores crear unha instantánea para o **Escenario 2.C - Backup**, para a *dserver00*, *uclient01* e *uclient02*.

-- Antonio de Andrés Lema e Carlos Carrión Álvarez --