

Xestión avanzada de Ubuntu. Servizos de rede e deseño dunha infraestrutura de dominio

Parte V

Obxectivos da parte V

Neste apartado veranse temas relacionados con:

- Cotas de disco.
- Programación de tarefas.
- Copias de seguridade

Ás cotas de disco

As cotas de disco permiten controlar o espazo de disco que consume cada usuario ou grupos de usuarios. As cotas para cada usuario son por sistema de ficheiros (particións) e non por carpetas, co cal se para carpetas distintas un usuario debera ter cotas distintas é necesario que cada unha delas resida nun sistema de ficheiros distinto (nun disco ou partición distinta).

Introdución

As cotas son implantadas polo **administrador do sistema** para cada sistema de ficheiros (disco/partición) para:

- **usuarios:** cada usuario terá os seus propios límites.
- **grupos:** tódolos usuarios dun grupo, en conxunto, teñen os límites do propio grupo. Se un usuario do grupo supera os límites da cota os demais usuarios do grupo tamén se verán afectados.

Hai dous tipos básicos de cota:

- **Cota de disco ou de bloque:** determina a cantidade de espazo en disco que pode ser usada por un usuario ou grupo nun sistema de ficheiros (disco/partición).
- **Cota de ficheiros ou de inode:** determina o número máximo de carpetas ou ficheiros que un usuario/grupo pode crear nun sistema de arquivos (disco/partición).

Ademais os administradores poden definir dous límites para cada un dos tipos de cota anteriores:

- **Cota suave, flexible / soft quota:** é un límite de advertencia, que lle informa ó usuario que se está aproximando á *hard quota*. Este límite ten un período de graza, que se verá a continuación.
- **Cota forte, ríxida / hard quota:** límite que o usuario/grupo non pode superar.

Finalmente para a **soft quota** existe un período de graza:

- **Grace period:** Cando un usuario/grupo supera a *soft quota*, este ten un período de graza (por defecto 7 días) para que volva a baixar dese límite antes que se lle limite totalmente como se dunha *hard quota* se tratase.

Pódense aplicar períodos de grazas xenéricas para tódolos usuarios/grupos dun sistema de arquivos (disco/partición) ou para un usuario/grupo concreto dese sistema de arquivos.

Nos seguintes títulos instálase o xestor de cotas e xestionáranse as cotas dun sistema de arquivos (**/home/iescalquera**) con comandos e doutro sistema de arquivos (*/*) con webmin.

O modo que ten o sistema para saber cantos recursos ou canto consumen por usuario é moi sinxelo, simplemente se analiza de que obxectos (ficheiros/carpetas) é **propietario** ese usuario.

Este é o formato que ten unha cota: os tres primeiros números están en KB.

```
Disk quotas for user xan (uid 10003):
  Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
  /dev/sda6   4       10    17    0      1      0     0     0
```

- Lenda:
 - **blocks**: espazo consumido en disco
 - **quota**: soft quota
 - **limit**: hard quota
 - **files**: número de arquivos/carpetas que o usuario ten nese sistema de arquivos (de cantos é dono).
 - **grace**: canto tempo lle resta ao usuario para poder estar por enriba da soft quota.

Instalación do xestor de cotas

Para instalar:

```
sudo aptitude install quota
```

Xestión de contas con comandos: /home/iescalquera

Aplicar cotas ó sistema de arquivos

Unha vez que se instalou o paquete hai que indicar en que sistema de arquivos se desexa aplicar a cota, neste caso no sistema de arquivos montado en **/home/iescalquera**. Ademais débese indicar se se van aplicar cotas de usuario, de grupo ou ambas.

Editar o ficheiro **/etc/fstab** e engadir os parámetros que indican que se desexan aplicar cotas de usuario e grupo sobre o sistema de arquivos montado en **/home/iescalquera**.

```
sudo nano /etc/fstab
```

Olo cada quen ten UUID distinto, por iso cada quen debe engadir manualmente os parámetros **usrquota** (cotas de usuario) e **grpquota** (cotas de grupo) se se van aplicar cotas ós grupos, senón non é preciso indicar ese parámetro. Neste caso só se van aplicar as de usuario aínda que se configurarán tamén as de grupo.

```
UUID=e162c2ac-dbc6-446b-9adf-09040428eb97 /home/iescalquera ext4 defaults,acl,usrquota,grpquota 0 0
```

Finalmente, só resta remontar o sistema de arquivos ó que se lle aplicaron as cotas (se o sistema de arquivos está en uso hai que reiniciar):

```
sudo mount -o remount /home/iescalquera
```

Executando o comando *mount*

```
mount
#debera amosar algo como o seguinte:
/dev/sda6 on /home/iescalquera type ext4 (rw,acl,usrquota,grpquota)
```

Crear os ficheiros de cotas

Na raíz do sistema de arquivos no que se aplican as cotas deben existir dous arquivos nos que se almacena a información sobre as cotas e consumos de cada usuario: **aquota.user** e **aquota.group**. Eses ficheiros poden crearse manualmente:

```
sudo touch /home/iescalquera/aquota.user /home/iescalquera/aquota.group
sudo chmod 600 /home/iescalquera/aquota.user /home/iescalquera/aquota.group
```

Pero farase facendo uso do comando **quotacheck**, que chequea, repara e crea, se non existen os arquivos anteriores.

```
sudo quotacheck -avug

# a: indica que se aplique o comando para tódolos sistemas de arquivos que teñen cotas.
# v: amosa información de saída do comando.
# u: chequea os ficheiros de cotas de usuarios
# g: chequea os ficheiros de cotas dos grupos
# para ver máis parámetros como sempre: quotachek --help.
# este comando tamén podería executarse co mesmo resultado como:
# quotacheck -vug /home/iescalquera
```

Este comando débese executar coas cotas desactivadas, que é como están a primeira vez, para desactivalas usar o comando **quotaoff**.

Finalmente activar as cotas:

```
quotaon -av

# a: activa as cotas para tódolos sistemas de arquivos que teñen cotas.
# v: amosa información de saída do comando.
# como sempre máis información en: quotaon -help.
# este comando tamén podería executarse co mesmo resultado como:
# quotaon -avug
# quotaon -v /home/iescalquera
```

Establecer cotas para un usuario

Para establecer cotas hai que facelo por usuario ou por grupo individualmente, para tratar varios usuarios débese realizar cun script ou combinación de comandos.

NOTA: para os usuarios que realizaron a tarefa 3 do curso de platega:
O usuario **xan** dos apuntes non foi tido en conta na tarefa, pero puido verse afectado pola pola aplicación de permisos. Para resolvelo:

- Introducir a xan no grupo **g-usuarios** (para que poida acceder a /home/iescalquera) e - sudo setfacl -Rm u:xan:rwX /home/iescalquera/xan

O mesmo acontece cos usuarios: alberto, alfredo, felipe, etc.

Para realizar o seguinte exemplo borráronse tódalas subcarpetas e ficheiros de /home/iescalquera/xan, salvo os tres arquivos que se copiaron de /etc/skel ó crear o usuario, cuxo propietario é root, neste exemplo. Ademais vaise entrar co usuario en modo consola e non en modo gráfico. A idea é comprobar como se ven afectados os límites de cada cota a medida que se crean carpetas e sabendo que estas ocupan 4KB.

Asignar unha cota a xan para o sistema de arquivos montado en /home/iescalquera con **setquota**.

```
sudo setquota -u xan 10 17 0 0 /home/iescalquera

# u: cota de usuario (g sería de grupo)
# 10: 13KB de soft quota de espazo de disco (bloques)
# 17: 17KB de hard quota de espazo de disco (bloques)
# 0: non hai soft quota para número de arquivos e subcarpetas (inodes)
# 0: non hai hard quota para número de arquivos e subcarpetas (inodes)
# neste caso como hai un só sistema de arquivos con cotas, o comando podería ser:
# sudo setquota -ua xan 10 17 0 0
```

```
# como sempre máis axuda en: setquota -help
```

Outra forma de editar as cotas dun usuario é con **edquota**:

```
sudo edquota -u xan
```

Ábrese un editor semellante ó *nano* que permite editar as cotas de usuario para cada sistema de ficheiros.

```
Disk quotas for user xan (uid 10003):
Filesystem          blocks      soft      hard      inodes     soft      hard
/dev/sda6            0           10        17         0           0         0

[ Léronse 3 liñas ]
^G Axuda      ^O Gravar    ^R Ler Fich  ^Y Páx Prev  ^K Cortar Txt ^C Pos Act
^X Saír      ^J Xustif   ^W Buscar    ^V Páx Seg   ^U Pegar Txt  ^T Ortografía
```

Como o usuario *xan*, neste caso, non ten ningún arquivo na súa carpeta persoal o seu consumo en espazo (**blocks**) e ficheiros/carpetas (**inodes**) é cero.

Os consumos pódense consultar cos seguintes comandos: **quota** e **repquota**

```
sudo quota xan
Disk quotas for user xan (uid 10003): none

# comando executado polo usuario root.
# no caso de que o propio usuario inicie sesión simplemente debe escribir: quota
```

O usuario *xan* aínda non ten nada no sistema de arquivos */home/iescalquera*, por iso indica *none*.

O resultado de executar **repquota**: informar das cotas dos usuarios en cada sistema de ficheiros.

```
sudo repquota -a
*** Report for user quotas on device /dev/sda6
Block grace time: 7days; Inode grace time: 7days

          Block limits              File limits
User      used  soft  hard  grace  used  soft  hard  grace
-----
root      --    1140    0    0      --    199    0    0
mon       --    1844    0    0      --    245    0    0
pia       --   17572    0    0      --    344    0    0
noe       --    1172    0    0      --    181    0    0
sol       --    2008    0    0      --    265    0    0
```

Este comando amosa os usuarios que son donos de algún obxecto no sistema de arquivos */dev/sda6*. Neste caso *xan* non é dono de ningún obxecto. Lembrar que o dono da súa carpeta e do seu contido (os tres ficheiros ocultos) é o *root*.

Observar tamén, como este comando indica que o período de graza para cada tipo de cota é, por defecto, de 7 días.

Estudo do uso de cotas

Vaise estudar como se modifican os consumos dun usuario a medida que este crea obxectos. Farase uso do usuario **xan**. Para iso iniciar a sesión, *en modo consola*, co citado usuario.

O usuario *xan*, comproba a súa quota, e non ten consumos. Lembrar que o dono da súa carpeta e do seu contido (os tres ficheiros ocultos) é o **root**.

```
xan@server00:~$ quota
Disk quotas for user xan (uid 10003): none
```

O usuario crea unha carpeta e comproba a súa quota.

```
xan@server00:~$ mkdir carpeta1
xan@server00:~$
xan@server00:~$ quota
Disk quotas for user xan (uid 10003):
  Filesystem  blocks   quota  limit  grace  files   quota  limit  grace
  /dev/sda6      4      10    17          1       0     0
```

Observar como a carpeta consume 4 KB (Blocks) e os ficheiros/carpeta creados é 1.

Por outra banda *root* pode consultar o consumo de tódolos usuarios: con **quota usuario** ou **repquota**.

```
administrador@server00:~$ sudo repquota -a

*** Report for user quotas on device /dev/sda6
Block grace time: 7days; Inode grace time: 7days

              Block limits                File limits
User          used  soft  hard  grace  used  soft  hard  grace
-----
root          --    1140    0    0          199    0    0
xan           --     4    10   17           1    0    0
mon          --   1844    0    0          245    0    0
pia          --  17572    0    0          344    0    0
noe          --   1172    0    0          181    0    0
sol          --   2008    0    0          265    0    0
```

Agora si que xa aparece o usuario *xan*.

O usuario *xan* crea 2 carpetas máis.

```
xan@server00:~$ mkdir carpeta2
xan@server00:~$ quota
Disk quotas for user xan (uid 10003):
  Filesystem  blocks   quota  limit  grace  files   quota  limit  grace
  /dev/sda6      8      10    17          2       0     0
xan@server00:~$
xan@server00:~$
xan@server00:~$ mkdir carpeta3
xan@server00:~$ quota
Disk quotas for user xan (uid 10003):
  Filesystem  blocks   quota  limit  grace  files   quota  limit  grace
  /dev/sda6    12*     10    17    7days    3       0     0
```

Unha vez superada *soft quota* actívase o período de graza, indicándolle ó usuario que lle quedan 7 días (ou os días ou horas ou minutos que restan en función de cando se consulten as cotas).

O usuario `root`, pasado un minuto despois de *xan* superar a *soft quota*, dispón da seguinte información:

```
administrador@server00:~$ sudo repquota -a
*** Report for user quotas on device /dev/sda6
Block grace time: 7days; Inode grace time: 7days
-----
User                Block limits          File limits
      used      soft  hard  grace      used  soft  hard  grace
-----
root      --      1140      0      0           199      0      0
alberto  --              8      0      0              3      0      0
xan      +-       12     10     17  6days         3      0      0
mon      --      1844      0      0           245      0      0
pia      --     17572      0      0           344      0      0
noe      --      1172      0      0           181      0      0
sol      --      2008      0      0           265      0      0
```

Fixarse que indica como período de graza 6 días no canto de 7, isto é porque xa quedan menos de 7 días exactos (que foi cando *xan* consultou as cotas xusto despois de superar a *soft quota*). Quedan 6 días con con varias horas e minutos. De feito se se executa *quota* no usuario *xan* xa indica o mesmo.

Mentres non pase o período de graza o usuario pode seguir creando obxectos ate chegar ó límite **hard quota**. Se pasa o período de graza o usuario non pode crear máis obxectos aínda que non chegara á **hard quota**.

Finalmente fixarse no + que hai na entrada de *xan*, indica que se superou a **soft quota** de bloques.

Xestión de períodos de graza

Os períodos de graza poden xestionarse para un usuario concreto ou para todo o sistema de arquivos.

- Xestionar o o período de graza de **todo o sistema de arquivos** con: **edquota -t** ou **setquota -t**.

```
sudo edquota -t

Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem          Block grace period    Inode grace period
/dev/sda6            7days                7days

[ Léronse 4 liñas ]
^G Axuda      ^O Gravar      ^R Ler Fich      ^Y Páx Prev      ^K Cortar Txt      ^C Pos Act
^X Saír      ^J Xustif      ^W Buscar      ^V Páx Seg      ^U Pegar Txt      ^T Ortografía
```

Aquí poden modificarse os períodos de graza para todo un sistema de arquivos.

```
sudo setquota -ta X Y

# X e Y son os segundos que se asignan ó período de graza de cada tipo de cota (bloques ou inodos)
# 7 días son 604800 segundos
# t: para indicar que se desexan modificar os tempos de graza para sistemas de arquivos.
# a: modificar os tempos de graza para tódolos sistemas de arquivos.
```

- Para **cambiar o período de graza dun usuario** usaranse os comandos anteriores co parámetro **-T**.

```
sudo edquota -Tu xan

Times to enforce softlimit for user xan (uid 10003):
Time units may be: days, hours, minutes, or seconds
Filesystem          block grace          inode grace
/dev/sda6           603422seconds       unset

[ Léronse 4 liñas ]

^G Axuda      ^O Gravar      ^R Ler Fich    ^Y Páx Prev    ^K Cortar Txt  ^C Pos Act
^X Saír      ^J Xustif     ^W Buscar     ^V Páx Seg    ^U Pegar Txt   ^T Ortografía
```

Indica os segundos (ou horas/minutos/días) que quedan de período de graza para o usuario *xan*. Tamén se poderían modificar eses valores para o usuario *xan*.

Con setquota:

```
administrador@server00:~$ sudo setquota -Tua xan 60 120
setquota: Not setting inode grace time on /dev/sda6 because softlimit is not exceeded.

administrador@server00:~$ sudo quota xan
Disk quotas for user xan (uid 10003):
Filesystem blocks quota limit grace files quota limit grace
/dev/sda6 12* 10 17 00:01 3 0 0

# T: modificar os tempos de graza dun usuario ou grupo.
# u: aplicar a usuarios.
# a: aplicar en tódolos sistema de arquivos.
# 60: segundos de graza para o espazo consumido en disco. Ou sexa, pasado un minuto o usuario xan
# non vai poder seguir creando obxectos.
# 120: segundos para o período de graza de inodes, non se aplicará porque non se superou o límite de
# soft quota de inodes.
# Se se desexa que non teña cota, nos segundos hai que indicar un cero (0).
```

Cando un usuario supere o seu tempo de graza non pode crear máis obxectos aínda que non alcanzara a *hard quota*. Só lle resta:

- Borrar obxectos para estar por debaixo da **soft quota**.
- Que un administrador lle amplíe o tempo de graza.

None indica que o usuario xa non ten ningún tempo de graza.

```
administrador@server00:~$ sudo quota xan
Disk quotas for user xan (uid 10003):
Filesystem blocks quota limit grace files quota limit grace
/dev/sda6 12* 10 17 none 3 0 0
```


Cotas excedidas e inicio de sesión gráfica

Unha vez que se alcance a **hard quota** ou se supere o *tempo de graza* da **soft quota** o usuario vai ter problemas co inicio de sesión gráfica, podendo non iniciarse a sesión, ó non poder crear tódalas carpetas e ficheiros que se crean/modifican cando se inicia sesión en modo gráfico.

Por exemplo, no caso anterior, no que se esgotou o tempo de graza para o usuario *xan*, este vai poder iniciar a sesión gráfica, pero non pode facer nada, pois *gnome* non pode crear tódalas carpetas e ficheiros de configuración personalizados do usuario *xan*.

Se algún proba a entrar en modo gráfico e ordenador se queda “colgado”, sen poder facer nada: entrar nunha consola (CTRL+F1) e reiniciar o entorno gráfico:

```
sudo /etc/init.d/gdm restart
```

Por tanto, para facilitar que **xan** poida iniciar sesión gráfica hai que ampliarlle a cota. Vaise establece unha cota de 100 MB e advertencia de aproximadamente 90 MB.

```
administrador@server00:~$ sudo setquota -u xan 90000 102400 0 0 /home/iescalquera.
```

Agora o usuario *xan* non terá problemas para iniciar a sesión en modo gráfico.

Copiar cotas

Para copiar as cotas dun usuario a outro úsase o comando **edquota -p**:

```
administrador@server00:~$ sudo edquota -p xan alberto
```

NOTA -LEMBRA: para os usuarios que realizaron a tarefa 3 do curso de platega:

O usuario **alberto** dos apuntes non foi tido en conta na tarefa, pero puido verse afectado pola aplicación de permisos da tarefa. Para resolvelo:

- Introducir a alberto no grupo **g-usuarios** (para que poida acceder a /home/iescalquera) e
- `sudo setfacl -Rm u:alberto:rwx /home/iescalquera/alberto`

Para comprobar a cota de *alberto*:

```
administrador@server00:~$ sudo quota alberto
```

Disk quotas for user alberto (uid 10002):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sda6	8	90000	102400		3	0	0	

O parámetro **-p** permite que a cota dun usuario prototipo se aplique a varios usuarios:

```
sudo edquota -p user_prototipo user1 user2 user3 ...
```

Aplicar cotas a varios usuarios do sistema

Hai veces nas que interesa aplicar unha cota a varios usuarios dun xeito sinxelo e sen ter que ir un por un.

Isto pódese realizar con: **setquota** ou **edquota**.

- Pasos previos a calquera dos dous comandos anteriores:

Primeiro hai que obter os usuarios ós que se lle vai aplicar as cotas:

No ficheiro **usuarios.completos** vaise almacenar o listado de tódolos usuarios que devolve *getent passwd*

```
getent passwd>usuarios.completos
```

No ficheiro **usuarios.nomes** almacenaranse só os nomes dos usuarios dos que se interese en aplicarlle as cotas.

```
echo `awk -F: '($3 > 9999) && ($3 < 10006) {print $1}' usuarios.completos`>usuarios.nomes
```

```
# awk -F: '($3 > 9999) && ($3 < 10006) {print $1}' usuarios.completos.
```

Esta liña quere dicir:

```
# awk é unha linguaxe de programación:
```

```
# 'awk -F: ---> indícalle a awk que o separador de campos é o signo ":".
```

```
# ($3 > 9999) && ($3 < 10006) {print $1} ----> indícalle a awk que se o campo número 3 (neste caso
#                               sería o campo de uid do arquivo usuarios.completos)
#                               está entre 9999 e 10006 entón que amose o campo 1
#                               dese ficheiro, isto é: o username.
```

```
# usuarios.completos ----> é o ficheiro que vai procesar awk para pasarllo ó comando echo e este
#                               envía ó ficheiro: usuarios.nomes
```

Agora en *usuarios.nomes* xa só se teñen os nomes dos usuarios ós que se lle desexa aplicar unha cota.

- Axustar cotas con **setquota**.

Crear o seguinte script: **setquotas.varios**

```
#!/bin/bash
for USUARIO in `cat usuarios.nomes`
do
    setquota -u $USUARIO 90000 102400 0 0 /home/iescalquera
done
```

Executalo:

```
sudo sh setquotas.varios
```

- Axustar cotas con **edquota**.

Con **edquota** é máis sinxelo xa que se lle poden pasar tódolos usuarios afectados como parámetro.

```
sudo edquota -p xan `cat usuarios.nomes`
# copia as cotas de xan ós usuarios que están no ficheiro usuarios.nomes
```

Tamén podería ser:

```
sudo edquota -p xan `awk -F: '($3 > 9999) && ($3 < 10006) {print $1}' usuarios.completos`
```

Quitar as cotas a un usuario

As cotas quíntanse poñendo zeros nos límites que se desexan desactivar.

```
sudo setquota -u USUARIO 0 0 0 0 /sistema de arquivos
```

As cotas de usuario nos equipos cliente

Os usuarios que inicien sesión nos equipos cliente, tanto GNU/Linux como MS Windows, xa teñen as cotas aplicadas de xeito transparente, ben por nfs ou ben por samba. As cotas en realidade son controladas polo servidor onde residen os sistemas de arquivos ós que os usuarios acceden remotamente.

As cotas en Ubuntu Desktop

En ubuntu desktop, *nautilus* (o xestor de arquivos) non informa de canto é o espazo real que o usuario ten nun sistema de arquivos montado por nfs, senón que amosa a capacidade total de todo o sistema de arquivos sen ter en conta a cota do usuario.

Por outra banda, en modo consola non se poden consultar as cotas.

En calquera dos dous casos, as cotas están funcionando, pois son controladas polo servidor de arquivos.

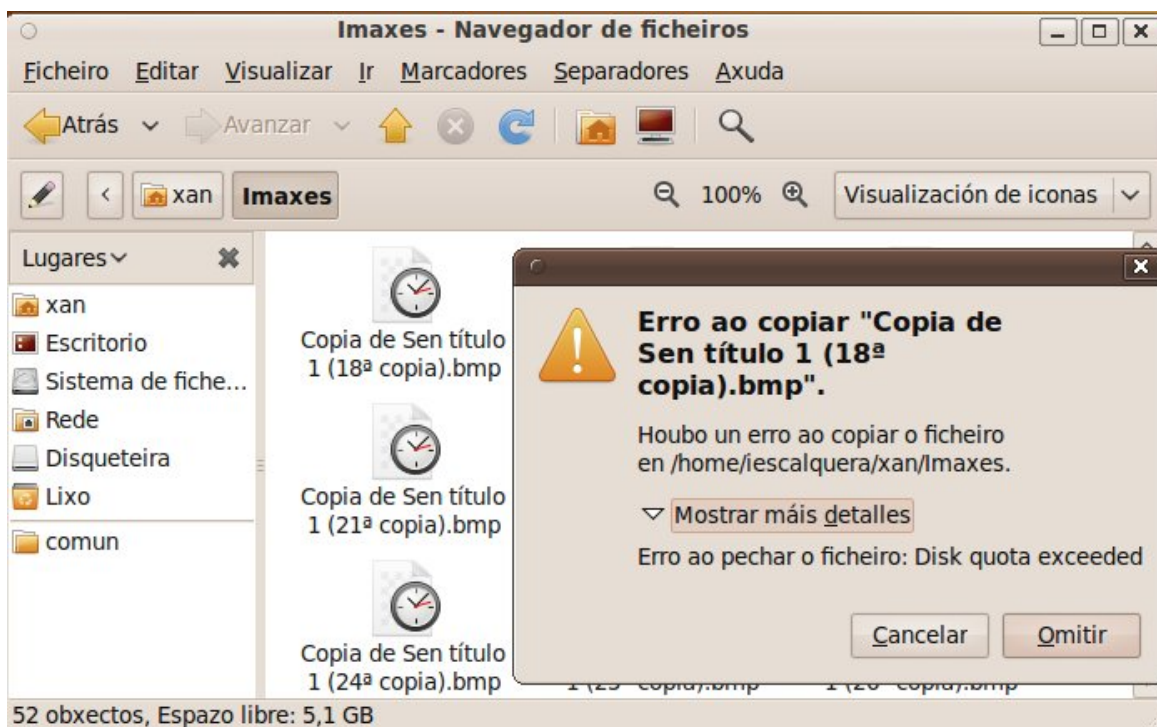
Nos postos de traballo para que os usuarios poidan consultar a súa cota en modo consola, precísase instalar o paquete *quota*.

```
sudo aptitude install quota.
```

No caso do navegador de arquivos *nautilus*, como xa se dixo, non informa de cal é o límite máximo de cota de disco para os usuarios, só informa cando o usuario non pode escribir no disco por superar a súa cota.

- **IMPORTANTE**, cando se borran arquivos e carpetas, estas van para á papeleira, estas seguen consumindo espazo no disco e por tanto na cota do usuario, haberá que baleirar o lixo se se desexa liberar espazo.

A seguinte imaxe amosa como informa *nautilus* dun exceso de cota para o usuario *xan*, pero en ningún caso lle indica que o espazo dispoñible para el é de 100 MB. Observar como aínda segue indicando que o espazo dispoñible son 5 GB e pico. Iso está informado no seguinte bug: <https://bugs.launchpad.net/ubuntuone-client/+bug/510719>



Ademais a aplicación das cotas con nautilus ten outro bug importante, como se relata en: <https://bugs.launchpad.net/ubuntu/+source/nautilus/+bug/410260>

Observar como xan consulta a súa cota, dende o cliente00:

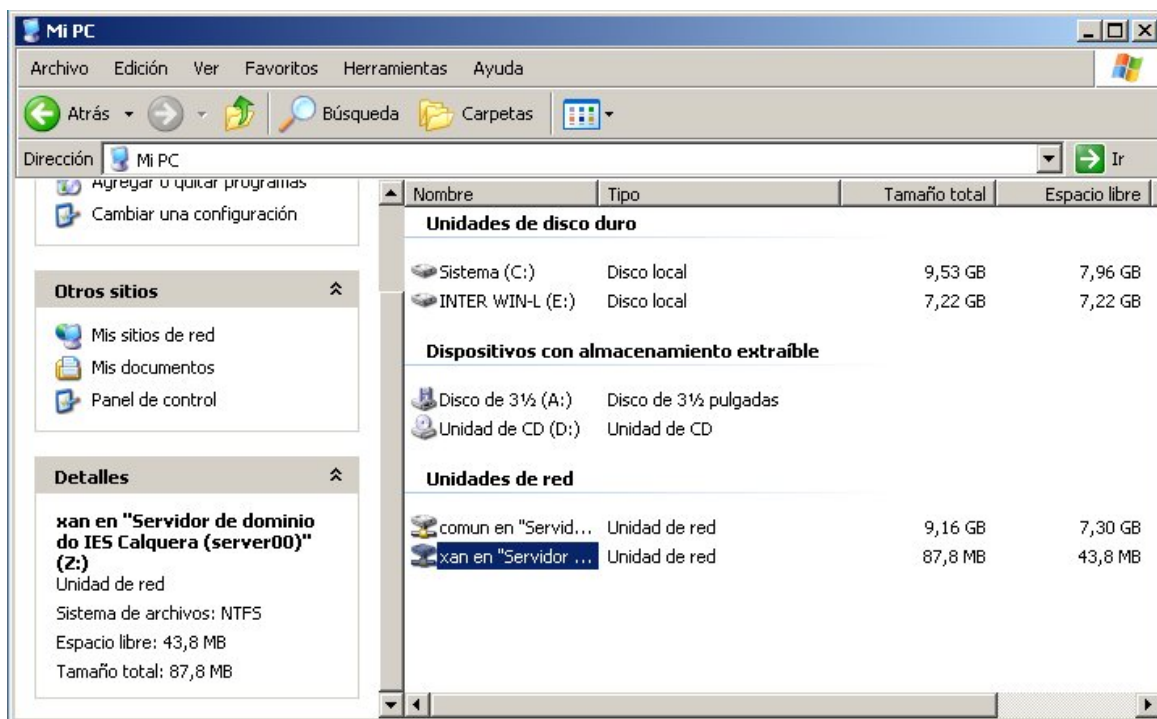
```
xan@cliente00:~$ quota
Disk quotas for user xan (uid 10003):
  Filesystem blocks quota limit grace files quota limit grace
server00.iescalquera.local:/iescalquera
                102400* 90000 102400 6days 763 0 0
```

Polo tamaño do ficheiro usado para realizar a proba para exceder a cota é imposible que o espazo consumido sume exactamente 102400 KB (a *hard quota*). Está explicado no bug anterior, e consiste que o último ficheiro que se copie antes de superar a *hard quota* vaise copiar integramente, cando se remate de copiar recortárase ata axustar o espazo que quedaba sen superar a cota. Ollo con iso en nautilus e co último arquivo que se copie antes de exceder a cota.

As cotas de usuario en Windows XP

Neste caso, MS Windows, si que *informa a medias* do espazo do que dispón o usuario na súa cota. Informa da *soft quota* e non da *hard quota*. Pero un usuario sempre terá este último límite como aplicable.

A seguinte imaxe amosa como informa Windows XP do espazo consumido e do espazo dispoñible en disco (**soft quota**). (90.000 KB = 87,8 MB) para o usuario *xan*.



O usuario *xan* poderá superar sen problemas eses 87,8 MB (sempre que non se supere o tempo de graza) ate 100 MB (hard quota).

Administración de cotas con webmin (sistema raíz, /)

As cotas de usuarios e grupos tamén poden ser administradas dende o webmin. Neste caso quérense xestionar as cotas para /comun. Pero resulta que /comun é unha simple carpeta no sistema de ficheiros montado na raíz /. Co cal, hai que xestionar as cotas do sistema de arquivos raíz. O ideal sería dispor dunha nova partición ou disco que estivera montado en /comun. Pero para realizar a práctica ben vale así, pero moito ollo coas cotas que se poñen no sistema raíz, pois poderíase bloquear o acceso ó sistema, coidado cos usuarios e grupos administradores. Só usar os usuarios creados no LDAP.

Antes de comezar actualizar os módulos dispoñibles en webmin: **Refresh Modules**

Activar cotas

En Sistema->Sistema de arquivo de disco e de rede Seleccionar o sistema raíz /, ó que se lle van activar as cotas, neste caso só de usuario, podería ser tamén de grupos. Premer en salvar.

En Sistema->Cotas de disco véñse os sistemas de arquivos sobre os que hai cotas. No caso do sistema raíz aínda non están activadas (mount -o remount /). Unha vez activadas, pódese comprobar con mount no servidor que o sistema raíz / ten o parámetro *usrquota*, e que o ficheiro *aquota.user* está creado na raíz /.

Xestionar cotas

Seleccionar o sistema de arquivos no que se desexa xestionar as cotas. Neste caso /.

Esta pantalla amosa aqueles usuarios que están usando espazo no sistema raíz. Se houbera usuarios con cotas, e sen obxecto tamén os amosaría. Para editar as cotas de cada usuario que xa é dono de obxecto e preciso premer sobre el. Para editar as cotas dun usuario que aínda non creou nada no sistema raíz, hai que seleccionalo explicitamente en **Editar Cota Para:**, neste caso o usuario *xan*.

Usuario	Estado	Limite por Usuario	Limite por Grupo	Cotas	Estado	Limite por Usuario	Limite por Grupo
xan	Activo	10240	Activo	10240	Activo	10240	Activo
root	Activo	10240	Activo	10240	Activo	10240	Activo
adm	Activo	10240	Activo	10240	Activo	10240	Activo
bin	Activo	10240	Activo	10240	Activo	10240	Activo
daemon	Activo	10240	Activo	10240	Activo	10240	Activo
ftp	Activo	10240	Activo	10240	Activo	10240	Activo
games	Activo	10240	Activo	10240	Activo	10240	Activo
irc	Activo	10240	Activo	10240	Activo	10240	Activo
lp	Activo	10240	Activo	10240	Activo	10240	Activo
mail	Activo	10240	Activo	10240	Activo	10240	Activo
man	Activo	10240	Activo	10240	Activo	10240	Activo
nfsnobody	Activo	10240	Activo	10240	Activo	10240	Activo
news	Activo	10240	Activo	10240	Activo	10240	Activo
operator	Activo	10240	Activo	10240	Activo	10240	Activo
postfix	Activo	10240	Activo	10240	Activo	10240	Activo
proxy	Activo	10240	Activo	10240	Activo	10240	Activo
rpc	Activo	10240	Activo	10240	Activo	10240	Activo
rpcuser	Activo	10240	Activo	10240	Activo	10240	Activo
sshd	Activo	10240	Activo	10240	Activo	10240	Activo
uucp	Activo	10240	Activo	10240	Activo	10240	Activo
xan	Activo	10240	Activo	10240	Activo	10240	Activo

A modo de exemplo asígnaselle unha cota de bloques de 100 MB para a *soft quota* e aproximadamente 150 MB para a *hard quota*. Premer en **actualizar**.

A pantalla amosa os usuarios que son donos de arquivos/ carpetas e aqueles que sen selo aínda teñen algunha cota. Premendo en **Editar tempos de gracia** pódense xestionar os tempos de graza do sistema de arquivo, non do usuario. Premendo en **Default quotas** webmin, permite asignar cotas ós usuarios que se van creando dende a propia ferramenta de creación de usuarios de webmin.

A modo de exemplo asígnase 100 MB de *soft quota* e aproximadamente 150 MB de *hard quota* para os futuros usuarios. Premer en **aplicar**.

Volvendo a **Sistema->Cotas de disco**, pódese editar as cotas que un usuario ten nos distintos sistemas de arquivos ...

... O usuario *xan* e as súas cotas. Premendo en **Copiar Cotas** pódense copiar as cotas do usuario *xan* a outros...

... neste caso ós usuarios *alfredo* e *alberto*.

Axustar cotas personalizadas en webmin

Puidera resultar interesante asignar unhas cotas por defecto a todo usuario salvo a aqueles que pertencesen a un determinado grupo. Por exemplo, sería interesante que cando se dese de alta un usuario no sistema este tivese unha cota salvo se fose un profesor que tería outra distinta.

Para iso pódese botar man do script que servía para dar de alta ós usuarios en kerberos despois de facelo no LDAP.

Pero antes débese desactivar en webmin que se asignen cotas por defecto ós usuarios novos: **Sistema->Cotas de disco->Default quotas**, poñer todos os valores a ilimitado.

Engadir ó contido do script `/opt/admin_kerberos_principals.sh` ...

```
#!/bin/bash
case "$USERADMIN_ACTION" in
"CREATE_USER") kadmin.local -q "addprinc -x dn=uid=$USERADMIN_USER,ou=usuarios,$
```

```
"MODIFY_USER") kadmin.local -q "cpw -pw $USERADMIN_PASS $USERADMIN_USER";;
"DELETE_USER") kadmin.local -q "delprinc -force $USERADMIN_USER";;
esac
```

o seguinte

```
#comprobar se o usuario que se dá de alta é un profesor
groups $USERADMIN_USER | grep profes >/dev/null
RETORNO=$?

#se é un profesor aplicarlle unha cota maior
#noutro caso unha cota por defecto.
if [ $RETORNO = 0 ] ; then
    setquota -u $USERADMIN_USER 200000 250000 0 0 /
else
    setquota -u $USERADMIN_USER 102400 150000 0 0 /
fi
```

-- Antonio de Andrés Lema e Carlos Carrión Álvarez -- (28 feb 2010).

Planificador de tarefas: cron

Para programar/planificar tarefas con certa periodicidade ou en certas datas en GNU/Linux úsase o servizo cron ^[2]. O nome de **cron** vén do grego Chronos ^[3].

Introdución

- **Cron** é un servizo que corre en segundo plano e que executa tarefas cada certo tempo. (/etc/init.d/cron)
- **Cron** le ficheiros chamados **crontab** (CRON TABLE), nos que se almacenan as tarefas a executar e cando se deben executar.
- Cada usuario pode ter o seu propio ficheiro **crontab**, onde pode especificar as accións que desexa levar a cabo e as veces que deberían ser executadas. Ollo!!! as accións que se especifiquen levaranse a acabo se o usuario ten permisos para realizalas e se ten permitido ter un ficheiro **crontab**, que por defecto todo usuario ten dereito.
 - Se o usuario non está logueado realizaranse igualmente as tarefas especificadas no seu ficheiro **crontab**.
 - Os ficheiros **crontab** que cree cada usuario almacénase en **/var/spool/cron/crontabs/<nome de usuario>**.
- Existe, tamén, un ficheiro **crontab do sistema**, este almacénase en **/etc/crontab**. Nel, ó igual que nos outros, indícanse as tarefas a realizar, cando executalas *'usuario* que as debe executar.
- **Os usuarios que poden, ou non, executar tarefas programadas están nas seguintes listas:**
 - **/etc/cron.allow** (permitir)
 - **/etc/cron.deny** (denegar)



Chronos esperando no Cemiterio Monumental
Xenovés de Staglieno ^[1].

- Estes dous ficheiros non existen inicialmente nun sistema estándar Debian/Ubuntu, por tanto todo usuario pode executar tarefas programadas. Tan pronto como exista un deses ficheiro aplícase a permisión/denegación:
 - Existe `/etc/cron.allow` e non `/etc/cron.deny`: só poden executar cron os usuarios que estean no ficheiro `cron.allow`.
 - Existe `/etc/cron.deny` e non `/etc/cron.allow`: só poden executar cron os usuarios que NON estean no ficheiro `cron.deny`.
 - Non existe ningún: todo usuario pode executar cron.
- No directorio `/etc/cron.d/` pódese almacenar ficheiros tipo `crontab` co mesmo formato quen ten o ficheiro `/etc/crontab`. É máis aconsellable crear `crontabs` en `/etc/cron.d/` que modificar o ficheiro `/etc/crontab`, porque este pode ser modificado cando se realicen actualizacións de cron.
- Se está instalada a utilidade **anacron** ^[4] executaranse tódalas tarefas almacenadas en: `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` e `/etc/cron.monthly`. A diferenza de **cron**, se o ordenador estivo a apagado, ó acender mira se algunha tarefa deses directorios non foi executada no seu momento e procede con ela.
- O servizo de `cron` cada minuto revisa o ficheiro `/etc/crontab` e o contido das carpetas `/var/spool/cron/crontabs`, `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, `/etc/cron.monthly` e `/etc/cron.d` e se hai cambios, estes son cargados en memoria. Por iso, non é preciso reiniciar o servizo `decron` cada vez que se modifique un ficheiro `crontab`.
- Finalmente para administrar a planificación de tarefas tense:
 - comandos: `crontab`
 - web: `webmin`
 - gráfico: paquete `gnome-schedule` (Aplicativos --> Ferramentas do sistema --> Administrador de Tarefas)

Axuda

Pódese atopar axuda en:

- `man cron`
- `man crontab` (axuda do comando).
- `man 5 crontab` (axuda do formato do ficheiro).

Estrutura dos ficheiros `crontab`

- Cada ficheiro `'crontab'`, ben do usuario ou ben do sistema, ten a maior parte das veces unha estrutura interna como a que segue:

```
# m h dom mon dow user  command
m:   minutos           0-59
h:   hora              0-23
dom: día do mes       1-31
mon: mes              1-12 (ou 3 primeiras letras do nome)
dow: día da semana:   0-7 (0 ou 7 é Domingo, ou 3 primeiras letras
do nome)
command: comando a executar.

user: usuario que executa o comando: non existe nos crontab de usuario

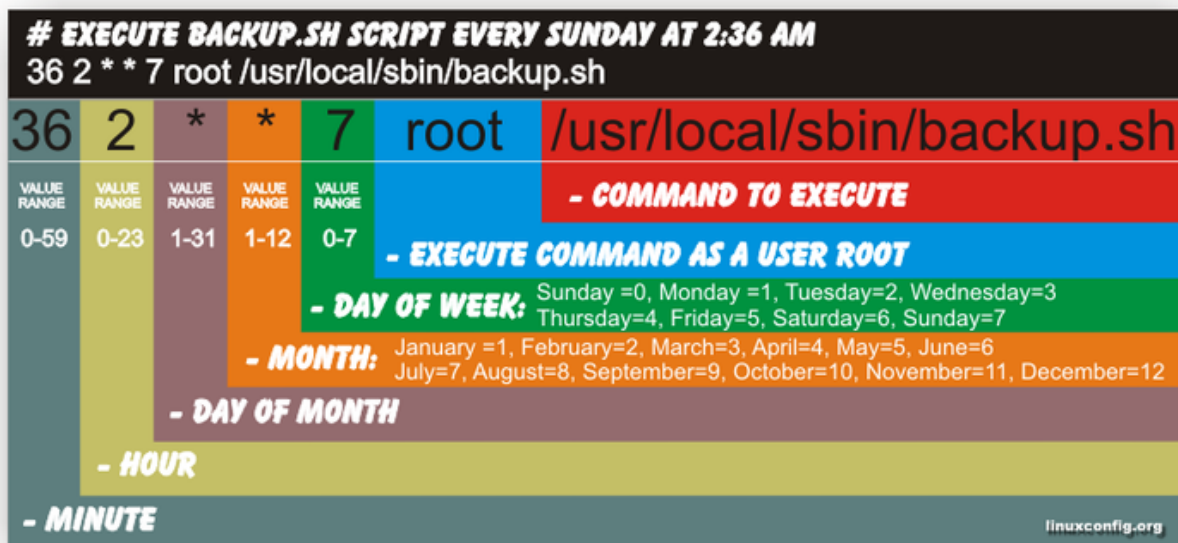
(/var/spool/cron/crontabs/<nome usuario>).
```

Si existe no **crontab do**

sistema

(/etc/crontab).

A seguinte imaxe pode clarificar un chisco máis a estrutura e os posibles valores:



- Esta imaxe ^[5] foi tomada de <http://www.linuxconfig.org>. Existe unha versión en español en Wiki Ballano ^[6].
- Neste caso, esta sería a configuración do **crontab do sistema** (/etc/crontab), pois ten o campo **usuario**.
- Notar que cando se indica un asterisco (*) e campo toma tódolos valores do rango, do primeiro ó último.
- Están permitidas listas (usar comas ,), rangos (usar guións -) ou saltos (usar barra /) de números en cada un dos campos. Exemplos para o campo minutos:
 - **1,5,7,20**: a tarefa executarase neses minutos.
 - **0-5**: a tarefa executarase cada minuto, dende o minuto 0 ate o 5, ambos inclusive.
 - **0-6, 10-13,40**: combinación dos 2 anteriores.
 - ***/15**: a tarefa executarase cada 15 minutos.
 - **0,15,30,45**: fai o mesmo que o anterior.
 - **0-11/2**: a tarefa executarase cada 2 minutos dende o minuto 0 ó 10 (incluídos).
 - **0,2,4,6,8,10**: fai o mesmo que anterior liña.
- Observar que o campo día está determinado por dous campos: dom (día do mes) e dow (día da semana). Se os dous teñen valor, o comando executarase cando CALQUERA dos campos coincidan co momento correcto.

Exemplo:

```
# m h dom mon dow command
30 4 1,15 * 5 comando
```

O comando executárase ás 4:30 am, os días 1 e 15 de cada mes ou calquera venres á mesma hora.

- No canto dos cinco primeiros campos pode aparecer unha das 8 cadeas especiais:

cadea	significado	m	h	dom	mon	dow
@reboot	Executar unha vez ó iniciar.					
@yearly	Executar unha vez ó ano, o 01/01/ano.=	"0	0	1	1	*"
@annually	O mesmo que @yearly.					
@monthly	Executar unha vez ó mes, o 01/mes/ano.=	"0	0	1	*	*"

@weekly	Executar unha vez á semana, o Domingo.=	"0 0 * * 0".
@daily	Executar unha vez ó día, ás 12 da noite.=	"0 0 * * *".
@midnight	O mesmo que @daily.	
@hourly	Executar cada hora en punto.=	"0 * * * *".

Envío de correo

- **cron** envía ó usuario dono da tarefa (root@localhost, usuario@localhost) que se executa programadamente un correo coa saída da tarefa ou dos comandos da tarefa, se teñen. Nun sistema Ubuntu recién instalado non hai ningunha utilidade de mail de consola, por tanto *cron* non vai poder enviar correo algún.
- Se se tivera instalada unha utilidade de correo, *cron* enviaría, por defecto, un correo ó dono da tarefa, coa saída desta, ou a quen se especifique no parámetro **MAILTO** do ficheiro *crontab*.
- Que facer no caso de ter unha utilidade de mail e non se desexa que cron estea enviando unha mensaxe de correo coa saída da tarefa vez que se executa?:
 - **comando &> /dev/null** ou **comando > /dev/null 2>&1**: envía a saída do comando/erros a /dev/null.
 - **comando >> ruta/saida.txt 2>&1**: envía a saída do comando/erros ó ficheiro *saida.txt*
 - **MAILTO=""**: definir o parámetro do ficheiro **crontab** a nada, e enviaráselle o correo á **ninguén**.

No caso que nos ocupa non hai utilidade de mail instalada. Pero verase un exemplo máis adiante con isto.

Variables de entorno

- Algunhas variables de entorno son iniciadas polo servizo *cron*. Cando se lanza unha tarefa programada estes son os seus valores por defecto.
 - HOME= colle o valor do ficheiro */etc/passwd* do dono do *crontab*.
 - LOGNAME= colle o valor do ficheiro */etc/passwd* do dono do *crontab*.
 - PATH=/usr/bin:/bin
 - SHELL=/bin/sh

Estes valores poden ser modificados ó inicio dun ficheiro *crontab*. E poden ser definidas outras novas variables, como: LANG, que por defecto é inglés. Se se desexase que a locale estivese en galego, indicar ó inicio do *crontab*:

```
LANG=gl_ES.UTF-8
```

Exemplos de ficheiros crontab

Antes de entrar en materia e realizar tarefas programadas véxanse algúns exemplos destas. Ó final destes exemplos creranse varias tarefas programadas.

- **Exemplo de Ficheiro crontab dun usuario: /var/spool/cron/crontabs/<usuario>**:

```
# Un ficheiro de usuario debe ser editado co comando crontab, logo verase.

#Usar /bin/bash para executar os comandos, no canto do shel por defecto: /bin/sh
SHELL=/bin/bash
#
# Enviar un correo, se estivera configurado, coa saída dos comandos a tres
# usuarios, o último é un usuario local. Non importa que sexa o dono do crontab
MAILTO=usuario@edu.xunta.es, user@gmail.com, usuario_local

# Executa o script "diario.job", tódolos días, cada 5 minutos, despois da media
# noite. A súa saída non é enviada ó correo senón que se envía ó ficheiro
```

```

# saida.txt
5 0 * * *          $HOME/bin/diario.job >> $HOME/saida.txt 2>&1

# Execútase a tarefa monthly ás 2:15pm do primeiro de cada mes, e envíase un
# correo coa súa saída ós tres destinatarios de MAILTO.
15 14 1 * *       $HOME/bin/monthly

#
# Enviar un corre a xose as 10 pm ós días de semana:
0 22 * * 1-5      mail -s "Son as 10pm" xose%Xose,%%Onde están os nenos?%

# Envía a saída do comando echo ós tres de MAILTO, tódolos días, cada 2 horas,
# 23 minutos despois da hora.
23 0-23/2 * * *   echo "run 23 minutes after midn, 2am, 4am ..., everyday"

# Envía a saída do comando echo ós tres de MAILTO, cada domingo ás 4:05 am.
# 23 minutos despois da hora.
5 4 * * sun       echo "run at 5 after 4 every sunday"

# Envía o erro da saída do comando mkdir /u ós tres de MAILTO, cada minuto.
# Un usuario calquera non ten, por defecto, permisos para crear na raíz.
# A mensaxe que enviaría sería algo como isto:
#          mkdir: non se pode crear o directorio "/u": Permission denied
* * * * *         mkdir /u

```

- **Exemplo de Ficheiro crontab de sistema: /etc/crontab:**

```

# Este ficheiro, ó igual que os crontabs que se creen dentro de /etc/cron.d/,
# ten un campo de usuario, que indica que usuario executa ese comando.

# Este ficheiro óu os crontabs de /etc/cron.d/ editanse á man sen usar o
# comando crontab, como se verá máis adiante.

# Non se recomenda que se modifique o ficheiro /etc/crontab, pois se se
# actualiza o cron, pode ser que se actualice este ficheiro. Para tarefas
# programadas do sistema créense ficheiros crontabs en /etc/cron.d/

# Este é o contido real de /etc/crontab.
#Definición de variables de entorno.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# run-parts: executa programas ou scripts dun directorio.
# m h dom mon dow user      command

17 * * * * root    cd / && run-parts --report /etc/cron.hourly

# Comproba se existe o comando anacron run
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )

```

```

47 6      * * 7      root      test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6      1 * *      root      test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )

# A modo de exemplo vaise engadir unha entrada neste ficheiro, cousa non
# aconsellable, é mellor crear un ficheiro tipo crontab en /etc/cron.d/

MAILTO=usuario@edu.xunta.es, user@gmail.com, usuario_local

# Envía a saída do comando (o espazo que consumen as carpetas persoais dos
# usuarios LDAP) ós tres de MAILTO. Ás 2am de cada día.
* 2 * * * root du -h --max-depth=1 /home/iescalquera

```

Crear tarefas programadas: liña comandos

Vistos os exemplos anteriores vanse realizar varias tarefas programadas no server00.

Tarefas programadas de usuarios

Usar o comando **crontab**:

```

crontab [-u usuario] { -e | -l | -r }

# Parámetros
# -u: para editar o crontab dun usuario distinto ó actual.
# -e: edita o ficheiro crontab de usuario
# -l: lista o ficheiro crontab de usuario
# -r: borra o ficheiro crontab de usuarios

```

- **A miña primeira tarefa programada:**

Crear unha tarefa programada que cada minuto escriba nun ficheiro o nome do usuario, o seu *home*, e a data completa.

Editamos o ficheiro crontab do usuario *administrador*. Non hai en */var/spool/cron/crontabs* ningún ficheiro crontab para o administrador e ademais a primeira vez que se executa o comando crontab hai que indicarlle que editor de texto se desexa usar, neste caso úsase *nano*:

```

administrador@server00:~$ crontab -e
no crontab for administrador - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano      <---- easiest
 3. /usr/bin/vim.basic
 4. /usr/bin/vim.tiny

Choose 1-4 [2]:

```

Na tarefa programada introducir:

```

# m h dom mon dow  command
* * * * * echo "$LOGNAME, $HOME " `date`>>ficheiro.txt

```

- Notar que non ó ficheiro non se lle puxo a ruta, a ruta que tomará por defecto é o home do usuario. Pois esa tarefa execútase no contorno do usuario administrador. Sería equivalente a escribir *\$HOME/ficheiro.txt*.
- En **/var/spool/cron/crontabs/administrador** foi onde se gardou o contido da tarefa programada do usuario administrador:

```
administrador@server00:~$ sudo ls /var/spool/cron/crontabs/ -l
total 8
-rw----- 1 administrador crontab 321 2010-03-21 12:29 administrador
-rw----- 1 root          crontab 230 2010-03-21 12:29 root
```

- Pasados uns minutos:

```
administrador@server00:~$ cat ficheiro.txt
administrador, /home/administrador Sun Feb 21 12:30:27 CET 2010
administrador, /home/administrador Sun Feb 21 12:31:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:32:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:33:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:34:01 CET 2010

administrador@server00:~$ date
Dom Feb 21 12:34:18 CET 2010
```

- Notar como o comando `date` executado na tarefa programada pon o día en inglés pero a saída de executar o comando pona en galego.
- Configurar a variable de entorno na tarefa programada.

```
administrador@server00:~$ crontab -e
```

Engadir: `LANG=gl_ES.UTF-8`, revisar o comando locale.

```
LANG=gl_ES.UTF-8

# m h dom mon dow    command
* * * * * echo "$LOGNAME, $HOME " `date`>>ficheiro.txt
```

- Pasados uns minutos, observar a última liña.

```
administrador@server00:~$ cat ficheiro.txt
administrador, /home/administrador Sun Feb 21 12:30:27 CET 2010
administrador, /home/administrador Sun Feb 21 12:31:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:32:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:33:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:34:01 CET 2010
administrador, /home/administrador Sun Feb 21 12:35:01 CET 2010
administrador, /home/administrador Dom Feb 21 12:36:01 CET 2010
```

- **A segunda tarefa programada: script**

Trátase de facer o mesmo que no caso anterior, pero que o comando a executar sexa un script:

```
nano diahora.sh
```

- Editar o seu contido

```
#!/bin/bash
echo "$LOGNAME, $HOME " `date`>>ficheiro2.txt
```

- Editar o crontab de administrador e modificar o seu contido:

```
crontab -e
```

- Editar contido:

```
LANG=gl_ES.UTF-8
```

```
# m h dom mon dow  command
* * * * * sh diahora.sh # Segunda tarefa
```

- Pasados uns minutos:

```
administrador@server00:~$ cat ficheiro2.txt
administrador, /home/administrador Dom Feb 21 12:55:01 CET 2010
administrador, /home/administrador Dom Feb 21 12:56:01 CET 2010
```

- **Terceira tarefa programada**

Quérese recibir un correo cada minuto coa mesma información anterior e ademais co contido de /root. Como non hai instalada unha utilidade de mail, vaise redireccionar a saída a un ficheiro mail.txt.

- Crear un script:

```
nano diahora_mail.sh
```

- Contido:

```
#!/bin/bash
echo "$LOGNAME, $HOME " `date`
ls /root
```

- Observar que as saídas dos comandos non se redireccionan a ningures. Ademais o usuario administrador non ten permisos para ver o contido de /root.
- Editar o crontab de administrador e modificar o seu contido:

```
crontab -e
```

- Editar contido:

```
LANG=gl_ES.UTF-8
```

```
# m h dom mon dow  command
* * * * * sh diahora_mail.sh >> mail.txt 2>&1 # Terceira tarefa
```

```
# Observar que a saída que resulta de executar o script mándase a mail.txt.
```

- Pasados outros minutos:

```
administrador@server00:~$ cat mail.txt
administrador, /home/administrador Dom Feb 21 13:10:02 CET 2010
ls: non se pode abrir o directorio /root: Permission denied
administrador, /home/administrador Dom Feb 21 13:11:02 CET 2010
ls: non se pode abrir o directorio /root: Permission denied
```

- Observar como amosa a saída dos dous comandos:
 - echo, saída correcta
 - ls /root: saída con erros, porque o usuario non pode acceder a /root.
- O usuario administrador xa tería recibidos 2 correos se non se redireccionara a saída.

Editar tarefas programadas doutros usuarios

Só o pode facer o administrador do sistema (root ou pertencentes a sudoers).

```
administrador@server00:~$ sudo crontab -u xan -e
administrador@server00:~$ sudo ls /var/spool/cron/crontabs/ -l
-rw----- 1 administrador crontab 330 2010-03-21 13:08 administrador
-rw----- 1 root          crontab 230 2010-03-21 12:29 root
-rw----- 1 xan          crontab 243 2010-03-21 13:19 xan
```

Borrar tarefas programadas de usuarios=

```
administrador@server00:~$ sudo crontab -u xan -r
```

Tarefas programadas de sistema

Estas poden ser creadas cun editor calquera, modificando o ficheiro **/etc/crontab** ou creando un ficheiro crontab no directorio **/etc/cron.d/**. Isto último é o aconsellable. Lembrar que hai que especificar o usuario.

Se se desexa que se execute un script, este debe residir nun sitio accesible para o usuario que o executa.

Exemplo de entradas en **/etc/crontab** ou **/etc/cron.d/tarefa**

```
sudo nano /etc/cron.d/tarefa
```

- O contido

```
# m h dom mon dow user      command
* * * * * administrador echo ola>>ficheiro4.txt
30 23 * * * root          /var/local/sbin/backup.sh
@reboot          root          ntpdate es.pool.ntp.org

#POR FIN, xa se pode actualizar a hora do servidor, ó inicialo.
```

Tareas programadas: webmin

O administrador de tareas de webmin está n menú **Sistema-->Tareas Planificadas**.

A seguinte imaxe amosa tarefas do sistema e do sistema.

Configuración de Módulo

Tareas de Cron Planificadas

Find Cron jobs matching Search

Seleccionar todo. | Invertir selección. | Crear una nueva tarea de cron en catálogo | Crear una nueva variable de entorno. | Controlar el acceso de usuarios a tareas de cron

Usuario	¿Activa?	Comando	Mover
<input type="checkbox"/> root	Si	/etc/cron.daily/man-db /etc/cron.daily/apt /etc/cron.daily/aptitude /etc/cron.daily/standard /etc/cron.daily/samba /etc/cron.daily/appport /etc/cron.daily/popularity-contest /etc/cron.daily/bsdmainutils /etc/cron.daily/apache2 /etc/cron.daily/mlocate /etc/cron.daily/logrotate /etc/cron.daily/dpkg	
<input type="checkbox"/> root	Si	/etc/cron.weekly/man-db	
<input type="checkbox"/> root	Si	/etc/cron.monthly/standard	
<input type="checkbox"/> root	Si	/var/local/sbin/backup.sh	
<input type="checkbox"/> root	Si	ntpdate es.pool.ntp.org	
<input type="checkbox"/> root	Si	[-x /usr/lib/php5/maxlifetime] && [-d /var/lib/php5] && find /var/lib/php5/ ...	
<input type="checkbox"/> administrador	Si	echo ola>>ficheiro4.txt	
<input type="checkbox"/> administrador	Si	Variable de entorno LANG = gl_ES.UTF-8	↓
<input type="checkbox"/> administrador	Si	sh diahora_mail.sh >> mail.txt 2>&1	↑

Seleccionar todo. | Invertir selección. | Crear una nueva tarea de cron en catálogo | Crear una nueva variable de entorno. | Controlar el acceso de usuarios a tareas de cron

Delete Selected Jobs Disable Selected Jobs Enable Selected Jobs

Premendo nas tarefas pódese acceder ó seu contido e planificación:

Índice de Módulo

Editar Tarea de Cron

Detalles de Tarea

Ejecutar tarea de cron como ...

¿Activa? Si No

Comando

Entrada del comando

Descripción

Cuándo ejecutar

Planificación simple .. Horariamente Horas y fechas seleccionadas abajo ..

Minutos	Horas	Días	Meses	Días de Semana
<input checked="" type="radio"/> Todos <input type="radio"/> Seleccionado... 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59	<input checked="" type="radio"/> Todos <input type="radio"/> Seleccionado... 0 1 2 3 4 5 6 7 8 9 10 11	<input checked="" type="radio"/> Todos <input type="radio"/> Seleccionado... 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	<input checked="" type="radio"/> Todos <input type="radio"/> Seleccionado... Enero Febrero Marzo Abril Mayo Junio Julio Agosto Setiembre Octubre Noviembre Diciembre	<input checked="" type="radio"/> Todos <input type="radio"/> Seleccionado... Domingo Lunes Martes Miércoles Jueves Viernes Sábado

Nota: Ctrl-click (ó comando-click en Mac) para seleccionar y deseleccionar minutos, horas, días y meses.

Rango de fechas en las que ejecutar

Ejecutar en cualquier fecha
 Sólo ejecutar desde /Ene / a /Ene /

Salvar Ejecutar Ahora Clone Job Borrar

As tarefas que se crean en webmin almacénase en crontabs en `/var/spool/cron/crontabs/<usuario>`, aínda que o usuario sexa o root.

-- Antonio de Andrés Lema e Carlos Carrión Álvarez -- (28 feb 2010).

Referencias

- [1] http://es.wikipedia.org/wiki/Cementerio_monumental_de_Staglieno
- [2] [http://es.wikipedia.org/wiki/Cron_\(Unix\)](http://es.wikipedia.org/wiki/Cron_(Unix))
- [3] <http://es.wikipedia.org/wiki/Chronos>
- [4] <http://packages.ubuntu.com/es/intrepid/admin/anacron>
- [5] http://www.linuxconfig.org/Linux_Cron_Guide
- [6] <http://wiki.ballano.net/index.php/Crontab>

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 Platega 08-09: Sistema operativo GNU-LINUX: UBUNTU 8.10.

Introdución

- De todos é sabido a importancia que ten que se realicen copias de seguridade ^[1] 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, aquí vanse usar dúas utilidades moi sinxelas, á par que potentes:
 - **tar** ^[2]: para empaquetar arquivos e comprimilos.
 - **rsync** ^[3]: 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:** cópanse tódolos arquivos.
- **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 diferenciales.
- **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 tódolos días, pero iso consume moito espazo e tempo para crear o backup, en cambio para restaurar, ante unha continxencia, sería o método máis rápido, 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 incremental, 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 o fin de semana. Os cambios para contemplar o fin de semana son moi intuitivos.

Simple

Unha copia total pola noite cada día hábil da semana. Cada copia dunha semana machaca a da 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.

- **Os pros:** moi sinxela

- **Os contras:** o backup histórico é moi corto.

Pai – Fillo (FS: Father - Son)

Unha copia incremental ou diferencial (mellor esta última) pola noite de luns a xoves e unha total ós 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 ó mes: 6
 - **Fillos:** diferenciais ou incrementais: copias 1, 2, 3 e 4
 - **Pai:** completas ou totais: copias 5 e 6
- **Os pros:** sinxela, xa se ten un histórico de 15 días atrás.
- **Os contras:** o backup histórico só é de 15 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 para o mes (9 en total) so 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, ós 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 ó mes: 9
 - **Fillos:** diferenciais ou incrementais diarias: copias 1, 2, 3 e 4
 - **Pai:** completas ou totais semais: 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 ralizar, o ideal 12 copias m.
- **Os contras:** ó 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.

Torres de Hanói

Baséase no xogo matemático das torres de Hanói ^[4]

- 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 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 sobre escrita 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		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

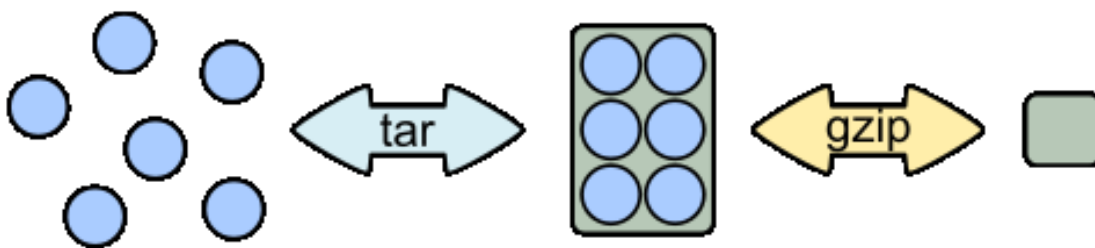
	Lun	Mar	Mér	Xov	Ven	Sáb	Dom
copia sem-1		A	B	A	C	A	-
copia sem-2		B	A	D	A	B	-
copia sem-3		A	C	A	B	A	-
copia sem-4		E	A	B	A	C	-
copia sem-5		A	B	A	D	A	-

- **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) ^[5] é unha utilidade que basicamente serve para empaquetar varios arquivos nun só arquivo contedor, opcionalmente pódense comprimir.

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



O comando tar non debe ser usado cando hai arquivos abertos ou en uso por programas.

Tipos de compresión

A parte de empacquetar 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 wiki ^[6] explícase o funcionamento de varios compresores libres.

Nesta outra faise unha comparativa ^[7] entre factor de compresión e tempo consumido dos algoritmos: gzip ^[8] (.gz), bzip ^[9] (.bz2) e LZMA ^[10] (.lzma). Este último é o que usa o programa 7-Zip ^[11].

Na páxina do comando tar ^[12] 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.

Formato do comando tar

O formato 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 cliente00 co usuario xan.

```
#Crear un empacuetado amosando o que fai
xan@cliente00:~$ tar cvf empacuetado1.tar dir1 dir2
dir1/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/
dir2/a1.txt
dir2/a2.txt

#Listar o contido dun empacuetado
xan@cliente00:~$ tar tf empacuetado1.tar dir1 dir2
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
xan@cliente00:~$ tar czf empaquetado2.tar.gz dir1 dir2

# Crear un tar e comprimilo autodetectando o formato pola extensión
xan@cliente00:~$ tar cfa empaquetado3.tar.lzma dir1 dir2

#Canto ocupa cada paquete?
xan@cliente00:~$ ls empaquetado* -lh
-rw-r--r-- 1 xan profes 20K 2010-03-22 09:53 empaquetado1.tar
-rw-r--r-- 1 xan profes 2,5K 2010-03-22 09:54 empaquetado2.tar.gz
-rw-r--r-- 1 xan profes 2,1K 2010-03-22 09:54 empaquetado3.tar.lzma

xan@cliente00:~$ rm dir1 -r
xan@cliente00:~$ rm dir2 -r
#Recuperar dir1 e dir2. As dúas sentencias seguintes son equivalentes.
xan@cliente00:~$ tar xfa empaquetado3.tar.lzma
xan@cliente00:~$ tar xf empaquetado3.tar.lzma

xan@cliente00:~$ rm dir1 -r

#Recuperar só dir1
xan@cliente00:~$ 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 usárase o parámetro: **--listed-incremental=ficheiro.snar** ou **-g**. O ficheiro *.snar é un arquivo que contén matadatos sobre os arquivos que están no empaquetado. Este ficheiro chámase **snapshot**, o seu obxectivo é axudar a determinar que ficheiros cambiaron, se engadiron ou borrarón 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, teñen sempre o mesmo UUID.

Copias Incrementais

- Para amosar o seu funcionamento vaise entrar en cliente00 co usuario *xan*, 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

xan@cliente00:~$ 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.
xan@cliente00:~$ nano dir1/f1.txt
xan@cliente00:~$ 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.
xan@cliente00:~$ 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.
xan@cliente00:~$ 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.
xan@cliente00:~$ 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.
xan@cliente00:~$ ls back* -lh
-rw-r--r-- 1 xan profes 316 2010-03-22 10:39 backup_inc_1.tar.bz2
-rw-r--r-- 1 xan profes 305 2010-03-22 10:40 backup_inc_2.tar.bz2
-rw-r--r-- 1 xan profes 211 2010-03-22 10:40 backup.snar
-rw-r--r-- 1 xan profes 2,6K 2010-03-22 10:38 backup_total.tar.bz2

#O contido de cada un dos tres arquivos comprimidos.
xan@cliente00:~$ tar tf backup_total.tar.bz2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

xan@cliente00:~$ tar tf backup_inc_1.tar.bz2
tar: Record size = 8 blocks
dir1/
dir2/
dir1/subdir/

dir1/f1.txt
xan@cliente00:~$ tar tf backup_inc_2.tar.bz2
tar: Record size = 16 blocks
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 que corresponde ó 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 10.10 é a 1.23
- 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
xan@cliente00:~$ rm backup.snar

#Copia total do segundo ciclo de copias
xan@cliente00:~$ tar cvfa backup_total.tar.bz2 -g backup.snar --no-check-device dir1 dir2
tar: dir1/subdir: O directorio é novo
dir1/
dir2/
dir1/subdir/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

#Arquivos implicados na copia
xan@cliente00:~$ ls backup* -lh
-rw-r--r-- 1 xan profes 316 2010-03-22 10:39 backup_inc_1.tar.bz2
-rw-r--r-- 1 xan profes 305 2010-03-22 10:40 backup_inc_2.tar.bz2
-rw-r--r-- 1 xan profes 211 2010-03-22 11:22 backup.snar
-rw-r--r-- 1 xan profes 2,7K 2010-03-22 11:22 backup_total.tar.bz2

#Contido da copia completa actual.
xan@cliente00:~$ tar tf backup_total.tar.bz2
dir1/
dir2/
dir1/subdir/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt
```

1. E agora tocaría 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
xan@cliente00:~$ 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
xan@cliente00:~$ nano dir1/f1.txt
xan@cliente00:~$ nano dir2/a3.txt
xan@cliente00:~$ mkdir dir1/subdir

#Copiar o arquivo de metadatos para primeira copia diferencial e usalo no tar.
xan@cliente00:~$ cp backup.snar backup_dif_1.snar
xan@cliente00:~$ 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/
dir2/
dir1/subdir/
dir1/f1.txt
dir2/a3.txt

#Modificar o ficheiro f2.txt
xan@cliente00:~$ 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 diferencial 1.
xan@cliente00:~$ cp backup.snar backup_dif_2.snar
xan@cliente00:~$ 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/
dir2/
dir1/subdir/
dir1/f1.txt
dir1/f2.txt
dir2/a3.txt

#Os arquivos implicados, fixarse como a cada diferencial o tamaño vai crecendo.
xan@cliente00:~$ ls back* -lh
-rw-r--r-- 1 xan profes 219 2010-03-22 11:54 backup_dif_1.snar
-rw-r--r-- 1 xan profes 388 2010-03-22 11:54 backup_dif_1.tar.bz2
-rw-r--r-- 1 xan profes 219 2010-03-22 11:55 backup_dif_2.snar
-rw-r--r-- 1 xan profes 609 2010-03-22 11:55 backup_dif_2.tar.bz2
-rw-r--r-- 1 xan profes 156 2010-03-22 11:52 backup.snar
-rw-r--r-- 1 xan profes 2,7K 2010-03-22 11:52 backup_total.tar.bz2

#O contido de cada arquivo da copia
xan@cliente00:~$ tar tf backup_total.tar.bz2
dir1/
dir2/
dir1/f1.txt
dir1/f2.txt
dir1/f3.txt
dir2/a1.txt
dir2/a2.txt

xan@cliente00:~$ tar tf backup_dif_1.tar.bz2
tar: Record size = 16 blocks
dir1/
dir2/
dir1/subdir/
dir1/f1.txt
dir2/a3.txt

xan@cliente00:~$ tar tf backup_dif_2.tar.bz2
tar: Record size = 16 blocks
dir1/
dir2/
dir1/subdir/
dir1/f1.txt
dir1/f2.txt
dir2/a3.txt
xan@cliente00:~$
```

- **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 autocumentado 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="/home /comun"    # 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=.bz2   # 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
# Cópíase o arquivo de metadatos, para poder realizar logo as copias
# diarias diferenciais.

tar cfa $F_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

    tar cfa $F_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.
```

```

cp $F_METADATOS_ULTIMA_TOTAL $F_METADATOS
tar cfa $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 (`/var/spool/cron/crontabs/root`)

```

LANG=gl_ES.UTF-8

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

```

- Para probar o script fixéronse os seguintes cambios no mesmo:

```

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

DIRECTORIOS="dir1 e dir2"    # Directorios a copiar
                              # Obrigatorio

```

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

A imaxe amosa o calendario do 2010.

```

administrador@server00:~$ cal 2010
                2010

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

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

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

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

```

O script comezouse a executar o 13 de marzo (un sábado, que segundo a configuración do script no 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 ó 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.

```

administrador@server00:~$ tree backup/10 -hND
backup
`-- [4.0K Mar 26 23:55] 2010
   |-- [2.6K Mar 13 23:55] fdc-00-Primeira-Sáb.tar.bz2
   |-- [4.7K Mar 26 23:55] fdc-01-Total-Mes-03-Mar.tar.bz2
   |-- [3.9K Mar 19 23:55] fdc-02-Total-Semana_3.tar.bz2
   |-- [3.3K Mar 29 23:55] fdc-03-Dif1-Lun.tar.bz2
   |-- [4.0K Mar 23 23:55] fdc-03-Dif2-Mar.tar.bz2
   |-- [4.1K Mar 24 23:55] fdc-03-Dif3-Mér.tar.bz2
   |-- [4.0K Mar 25 23:55] fdc-03-Dif4-Xov.tar.bz2
   `-- [4.0K Mar 26 23:55] snar
      |-- [ 766 Mar 26 23:55] fdc-00-ULTIMA_TOTAL.snar
      |-- [ 766 Mar 26 23:55] fdc-01-Total-Mes-03-Mar.snar
      |-- [ 395 Mar 19 23:55] fdc-02-Total-Semana_3.snar
      |-- [ 872 Mar 29 23:55] fdc-03-Dif1-Lun.snar
      |-- [ 607 Mar 23 23:55] fdc-03-Dif2-Mar.snar

```

```

|-- [ 660 Mar 24 23:55] fdc-03-Dif3-Mér.snar
|-- [ 713 Mar 25 23:55] fdc-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.
administrador@server00:~$ cat backup/2010/snar/rexistro.txt
Sáb Mar 13 23:55:00 CET 2010 Primeira:fdc - INICIO: Total. Sáb
Sáb Mar 13 23:55:00 CET 2010 Primeira:fdc - FIN      : Total.

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

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

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

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

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

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

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

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

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

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

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

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

Ven Mar 26 23:55:00 CET 2010 Avó      :fdc - INICIO: Total mensual Mar
Ven Mar 26 23:55:00 CET 2010 Avó-Pai :fdc - FIN      : Total

----- MES de Abril -----

```

```
Sáb Mar 27 23:55:00 CET 2010 Non      :fdc - Ós Sáb non se realizan backups

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

Rsync

rsync ^[3] é unha utilidade para manter sincronizados dous directorios ou arquivos. Mediante a técnica de Codificación Delta ^[13] permite realizar sincronizacións simplemente enviando ó 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.
```

Exemplo: sincronización local

Trátase de sincronizar unha carpeta con outra, as dúas no mesmo equipo. Fixarse como desactiva a transmisión delta.

```
#A carpeta2 non existe e será creada por rsync
administrador@server00:~$ rsync -azv 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.
administrador@server00:~$ nano carpeta1/f4
administrador@server00:~$ rm carpeta1/f6
```

```
#Volver a sincronizar, nesta ocasión coa opción de -delete.
administrador@server00:~$ rsync -azv --delete carpeta1 carpeta2
sending incremental file list
delta-transmission disabled for local transfer or --whole-file
carpetal/
deleting carpeta1/f6
carpetal/f4
carpetal/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.
administrador@server00:~$ rsync --delete -azv -e ssh carpeta1 administrador@localhost:/home/administrador/carpeta3
opening connection using: ssh -l administrador localhost rsync --server -vlogDtprze.iLs --delete ./home/administrador/carpeta3
administrador@localhost's password:
sending incremental file list
created directory /home/administrador/carpeta3
delta-transmission enabled
carpetal/
carpetal/f4
carpetal/f5
carpetal/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.
administrador@server00:~$ nano carpeta1/f4
administrador@server00:~$ rm carpeta1/f6

# Volver a sincronizar.
administrador@server00:~$ rsync --delete -azv -e ssh carpeta1 administrador@localhost:/home/administrador/carpeta3
opening connection using: ssh -l administrador localhost rsync --server -vlogDtprze.iLs --delete ./home/administrador/carpeta3
administrador@localhost's password:
sending incremental file list
delta-transmission enabled
carpetal/
deleting carpeta1/f6
carpetal/f4
```

```
carpetal/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
```

-- Antonio de Andrés Lema e Carlos Carrión Álvarez -- (28 feb 2010).

Referencias

- [1] http://es.wikipedia.org/wiki/Copia_de_seguridad
- [2] <http://es.wikipedia.org/wiki/Tar>
- [3] <http://es.wikipedia.org/wiki/Rsync>
- [4] http://es.wikipedia.org/wiki/Torres_de_Han%C3%B3i
- [5] <http://www.gnu.org/software/tar/manual/tar.html>
- [6] http://en.wikipedia.org/wiki/Category:Free_data_compression_software
- [7] <http://odzangba.wordpress.com/2009/03/25/gzip-vs-bzip2-vs-lzma/>
- [8] <http://es.wikipedia.org/wiki/Gzip>
- [9] <http://es.wikipedia.org/wiki/Bzip2>
- [10] <http://es.wikipedia.org/wiki/LZMA>
- [11] <http://gl.wikipedia.org/wiki/7-Zip>
- [12] <http://www.gnu.org/software/tar/manual/tar.html#SEC132>
- [13] http://es.wikipedia.org/wiki/Delta_encoding

Fuentes y contribuyentes del artículo

Objetivos da parte V *Fuente:* <http://informatica.iessancllemente.net/manuais/index.php?oldid=16379> *Contribuyentes:* Carrion

Ás cotas de disco *Fuente:* <http://informatica.iessancllemente.net/manuais/index.php?oldid=21398> *Contribuyentes:* Antonio, Carrion

Planificador de tarefas: cron *Fuente:* <http://informatica.iessancllemente.net/manuais/index.php?oldid=21419> *Contribuyentes:* Antonio, Carrion

Copias de seguridade: tar e rsync *Fuente:* <http://informatica.iessancllemente.net/manuais/index.php?oldid=21430> *Contribuyentes:* Carrion

