

Taller de Gestió del Sistema Operatiu

Notes diverses

Sebastià Vila

November 10, 2021

Contents

1	Treballar com superusuari	2
2	Usuaris i permisos	3
2.1	Alta d'un usuari	3
2.2	/etc/passwd i /etc/group	3
2.3	Modificació d'un compte	4
2.4	Esborrat d'usuaris i grups	4
2.5	Atributs especials d'un fitxer	4
3	Filesystem Hierarchy Standard	4
4	Distribucions	5
5	Paquets	6
5.1	Tipus de dependències	6
5.2	Tipus d'estat	7
5.3	Estructura d'un paquet	7
5.4	dpkg	8
6	Sistema de distribució de paquets. Repositoris	9
6.1	Configuració local del sistema de distribució	9
6.2	Subconfiguracions	10
6.3	Ordres principals	10
6.4	Paquets «automàtics»	11
6.5	Cerca de paquets	11

7 Gestió dels paquets	11
7.1 Actualització periòdica	11
7.2 Instal·lació de soft de tercers	11
7.3 Instal·lar software privatiu	12
7.4 Instal·lar altre soft lliure	12
8 Backup	13
8.1 Introducció	13
8.2 Terminologia i esquema	13
8.3 Els backups són perillosos	14
8.4 Els backups són cars	14
8.5 Cas simple 1	14
8.6 Disseny de l'estratègia	14
8.7 Calendari	14
8.8 Eina	15
8.9 Procediment de backup	15
8.10 Procediment de recuperació (a entrenar!!)	15

1 Treballar com superusuari

Hi ha un usuari privilegiat al que no afecten els permisos d'accés: superuser. És molt important ser conscients de que treballar com a superusuari és **perillós**:

- Per que un error pot provocar pèrdua de dades o interrupció del servei.
- Per que un malware de qualsevol tipus pot esdevenir molt més efectiu fent mal.
- Per que, sense adonar-se'n, es poden obrir portes que poden ser aprofitades per malware més endavant.

Lleis del superusuari:

1. Només la mínima feina imprescindible com a superusuari.
2. Mai executar software d'origen desconegut.
3. Pensar dues vegades abans de prémer return.

Per esdevenir superusuari:

`su [-]`

Sovint no es pot usar l'ordre anterior per que el superusuari està configurat sense accés.

Existeix una ordre que serveix per executar ordres sota la identitat del superusuari. Permet un cert control (si cal) de què es pot executar, a quines hores, etc. En general:

`sudo <ordre>`

També, per sessions de treball més llargues:

`sudo -i`

Es configura a través d'un fitxer de configuració `/etc/sudoers` que cal editar-lo d'una forma especial per garantir que no et quedaràs bloquejat 'sudoedit'. Normalment es configura de forma que si pertanys al grup 'sudo', tens permisos per usar sudo. Així evitem haver d'editar la configuració.

2 Usuaris i permisos

Compte d'usuari determina:

- La shell
- El password i grups
- El directori home
- Altres detalls

2.1 Alta d'un usuari

`adduser [compte]`

Crea compte sencer: usuari, home, etc. Mirar man page de l'ordre.

2.2 `/etc/passwd` i `/etc/group`

Les 'BD' dels comptes s'implementen sobre fitxers de text que tenen un format concret. Els camps es separen amb ':'. Tot i ser editables no és recomanable fer-ho.

2.3 Modificació d'un compte

```
usermod --expiredate 2014-10-30 sebas
usermod --append --groups dialout sebas
usermod --home /home/newhomesebas sebas
usermod --shell /bin/sh sebas
usermod [--lock|--unlock] sebas
```

2.4 Esborrat d'usuaris i grups

Els grups s'afegeixen automàticament quan se citen. També es poden esborrar:

```
groupdel grup
```

Es poden veure amb 'id'.

L'esborrat d'usuaris esborrant home i spool de correu:

```
userdel --remove user
```

2.5 Atributs especials d'un fitxer

Més metadades associades a un fitxer que determinen aspectes de la seva semàntica. N'hi ha molts i és interessant llegir la pàgina de man. Per exemple, el següent codi marca el fitxer com 'immutable' (no es pot alterar ni esborrar):

```
chattr +i fitxer
```

3 Filesystem Hierarchy Standard

Un estàndard del món Linux per organitzar els fitxers (del sistema) en directoris.

```
/
/bin
/boot
/dev
/etc
/home
/lib
/media
```

```
/mnt
/proc
/root
/run
/sbin
/tmp
/usr
/usr/bin
/usr/include
/usr/lib
/usr/local
/usr/local/bin
/usr/local/etc
/usr/local/lib
/usr/share
/var
```

4 Distribucions

Una distribució de Linux és una compilació de software organitzat adequadament per a poder funcionar i ser (distribuït) a tots els usuaris que ho necessitin.

Linux no es fabrica monolíticament. Cada peça està mantinguda i evolucionada per una comunitat de software lliure diferent, amb eines, llenguatges i cultures diferents. Una distribució va a 'pescar' en totes aquestes eines, les ensambla i ajusta per que puguin funcionar juntes i les prepara per tal que siguin distribuïbles.

A Debian:

<https://sources.debian.org/stats/> <https://wiki.debian.org/Statistics>

- 6209 a 9000 persones hi han col·laborat
- 59000 paquets
- 1,194,042,086 línies de codi (bullseye)
- 312,728,996 KB de codi en disc

Existeix tota una infraestructura + política per gestionar la distribució basada en paquets.

5 Paquets

Un paquet és un conjunt de fitxers que col·laboren entre ells per aportar una funció al sistema operatiu. Per exemple, emacs (codi, configuracions, man, helps, etc.).

Tot i que n'hi ha de fitxers font, aquí ens referirem als de fitxers compilats. A banda dels fitxers corresponents un paquet també conté:

- Metainformació
 - Descripcions
 - Dependències (Depen, Recomana, Suggereix, Conflicte, Substitueix, Trenca, Provisiona)
 - Dates i autors
 - Copyrights
- Scripts d'operació
 - Instal·lació
 - Configuració
 - Actualització
 - Desinstal·lació

El format dels paquets està ben establert i tenen sufix '.deb'.

Paquets virtuals ho són els que no instal·len res sinó que representen un element general 'www-browser' diversos paquets poden provisionar aquest element, per exemple 'firefox'.

5.1 Tipus de dependències

Package A depends on Package B if B absolutely must be installed in order to run A. In some cases, A depends not only on B, but on a version of B. In this case, the version dependency is usually a lower limit, in the sense that A depends on any version of B more recent than some specified version.

Package A recommends Package B, if the package maintainer judges that most users would not want A without also having the functionality provided by B.

Package A suggests Package B if B contains files that are related to (and usually enhance) the functionality of A.

Package A conflicts with Package B when A will not operate if B is installed on the system. Most often, conflicts are cases where A contains files which are an improvement over those in B. "Conflicts" are often combined with "replaces".

Package A replaces Package B when files installed by B are removed and (in some cases) over-written by files in A.

Package A breaks Package B when both packages cannot be simultaneously configured in a system. The package management system will refuse to install one if the other one is already installed and configured in the system.

Package A provides Package B when all of the files and functionality of B are incorporated into A. This mechanism provides a way for users with constrained disk space to get only that part of package A which they really need.

5.2 Tipus d'estat

Estats en que es pot trobar un paquet

unknown - the user has never indicated whether the package is wanted.

install - the user wants the package installed or upgraded.

remove - the user wants the package removed, but does not want to remove any existing configuration file.

purge - the user wants the package to be removed completely, including its configuration files.

hold - the user wants this package not to be processed, i.e. wants to keep the current version with the current status whatever that is.

Imatge del graf de dependències d'Ubuntu.

5.3 Estructura d'un paquet

Un paquet es una «compactació de tres tarfiles»

```
$ ar t dpkg_1.19.7_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
$ ar x dpkg_1.19.7_amd64.deb
$ ls
control.tar.gz data.tar.xz debian-binary dpkg_1.19.7_amd64.deb
$ tar tJf data.tar.xz | head -n 16
./
```

```
./
./etc/
./etc/alternatives/
./etc/alternatives/README
./etc/cron.daily/
./etc/cron.daily/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg
./etc/dpkg/dpkg.cfg.d/
./etc/logrotate.d/
./etc/logrotate.d/alternatives
./etc/logrotate.d/dpkg
./sbin/
./sbin/start-stop-daemon
./usr/
./usr/bin/
$ tar tJf control.tar.xz
./
./conffiles
./control
./md5sums
./postinst
./postrm
$ cat debian-binary
2.0
```

‘control’ es el fitxer que conté les metadades (instal·lació, dependències, etc.)

5.4 dpkg

<https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>

dpkg és l'eina bàsica de gestió dels paquets. Només treballa amb paquets locals.

```
# llista continguts d'un paquet instal·lat
dpkg -L paquet
# llista els paquets instal·lats que contenen tal fitxer
dpkg -S fitxer_pattern
# instala un paquet local
```



```
dpkg -i paquet
# acaba de configurar els paquets pendents
dpkg -configure -a
# mostra metadades d'un paquet
dpkg -I paquet
# mostra estat d'instal·lació d'un paquet instal·lat
dpkg -s paquet
# mostra el contingut d'un paquet
dpkg -c paquet
```

Evidentment hi ha una 'bd' dels paquets instal·lats en el sistema a '/var/lib/dpkg'. Per instal·lar un paquet cal que es compleixin totes les dependències, altrament no es pot instal·lar.

6 Sistema de distribució de paquets. Repositoris

A través d'un esquema mostrar l'arquitectura del sistema de distribució:

- repositoris a la xarxa (amb altres possibilitats: disc local, muntat, etc)
- repositori local
 - paquets local
 - index del repositori
 - bd de paquets instal·lats i el seu estat

També comentar el sistema d'instal·lació i com es relaciona amb el sistema de distribució de paquets.

6.1 Configuració local del sistema de distribució

Configuració primària a '/etc/apt/sources.list' tot i que poden haver-hi subconfiguracions. Potser explicar sobre la realitat l'estructura de les subconfiguracions.

Sintaxi:

```
deb url distribution component1 component2 component3 [...] componentX
```

Per exemple

```
deb https://deb.debian.org/debian buster main contrib non-free
```

Un fitxer de configuració clàssic podria ser aquest:

```
# Security updates
deb http://security.debian.org/ buster/updates main contrib non-free

## Debian mirror

# Base repository
deb https://deb.debian.org/debian buster main contrib non-free
# Stable updates
deb https://deb.debian.org/debian buster-updates main contrib non-free
# Stable backports
deb https://deb.debian.org/debian buster-backports main contrib non-free
```

6.2 Subconfiguracions

En directori `/etc/apt/sources.list.d/`. Se sumen a la principal i permet trencar en fitxers separats.

6.3 Ordres principals

Coneixen l'arquitectura de distribució i poden instal·lar i gestionar paquets fent tota la feina que calgui: descarregar, desplegar, configurar, etc. Es basen en la llibreria **apt** i n'hi ha una diversitat:

- `apt-get`
- `apt`
- `aptitude`
- `synaptic`

```
# actualitza l'index local del repositori
apt update
# instal·la
apt install paquet
# esborra
apt remove paquet
# resinstal·la però no les configuracions
apt --reinstall install postfix
# actualitza els paquets instal·lats
```

```
apt upgrade
# actualitza a una nova versió de la distribució
apt full-upgrade
```

També és interessant ‘aptitude’.

6.4 Paquets «automàtics»

Els paquets instal·lats automàticament per satisfer les necessitats d’un paquet que ens interessava. Tenen una marca especial que permet que, quan s’esborra el paquet d’interès, automàticament es puguin esborrar la resta. (dibuix)

```
apt autoremove
apt-mark auto paquet # marca el paquet com automàtic maualment
```

6.5 Cerca de paquets

Treballen amb la informació de l’índex local i coneixen tots els paquets de la distribució tant si estan instal·lats com no.

```
apt search paraula
apt show paquet # mostra la info del paquet
```

També eines basades en xapian (cal instal·lar ‘apt-xapian-index’ i executar ‘update-apt-xapian-index’ el primer cop.

```
axi-cache search mot
```

7 Gestió dels paquets

7.1 Actualització periòdica

7.2 Instal·lació de soft de tercers

Per què podem voler instal·lar soft que no és a la distribució?

1. Per que en volem una versió més nova
2. Per que és privatiu i no es pot distribuïr
3. Per que no és a la distribució tot i ser lliure

4. Per en volem una configuració/build molt especial o retocat

Tot i que el consell més important és **no fer-ho mai**, és possible fer-ho.

NB: En el món Micro\$oft (i altres privatius) no existeix el concepte 'distribució' i sempre cal instal·lar software de tercers. És substancialment diferent del món lliure.

En cada cas dels anteriors les estratègies a seguir són diferents:

7.3 Instal·lar software privatiu

En aquest cas mai el trobarem en una distribució però el propietari pot haver-lo empaquetat en format '.deb'. En aquest cas, simplement cal baixar els paquets que siguin i executar:

```
$ sudo apt install paquet.deb
```

Previsions:

- Assegurar-se que la font és fiable (pot introduir problemes de seguretat)
- Assegurar-se que està ben empaquetat i respecta les polítiques de instal·lació. Molts softwares de tercers tenen un empaquetat deficient.

També pot passar que el propietari mantingui un repositori propi que conté només el seu software. Aleshores, cal afegir-lo als nostres repositoris, potser afegir les claus i finalment actuar com en qualsevol altre paquet de la distribució.

Finalment podria ser que no estigui empaquetat i hi hagi una instal·lació específica. En aquest cas és altament recomanable fer una instal·lació local en comptes d'una instal·lació global per raons de seguretat i política de gestió.

7.4 Instal·lar altre soft lliure

A banda de les opcions anteriors i amb la mateixa consideració, disposem d'algunes eines que permeten instal·lar productes en formats empaquetats autocontinguts. Arrosseguen un excés de material però permeten instal·lacions independents del sistema operatiu. Són avantatjoses per fer proves amb softwares de versions molt recents, per exemple.

Un exemple és flatpak (<https://flatpak.org>). Per usar flatpak, cal instal·lar prèviament el paquet 'flatpak' i després demanar al gestor flatpak que instal·li el soft en que estem interessats:

```
$ flatpak install flathub com.spotify.Client  
$ flatpak run com.spotify.Client
```

8 Backup

8.1 Introducció

La tècnica de backup consisteix a obtenir una còpia precisa de l'estat d'un computador de manera que sigui reproduïble en un altre moment del temps.

Els backups es fan per moltes raons i cada raó diferent incideix en la manera o la tecnologia que es pot aplicar. És una operació molt més complexa que no sembla i, en un entorn professional, un element de moltíssima importància.

Algunes raons:

- Recuperació de fitxers esborrats o alterats
- Auditoria
- Reducció del risc davant problemes hardware (disc, etc.)
- Reducció del risc davant problemes de seguretat
- Reducció del risc davant operacions complexes d'actualització
- Arxivat de legacy

8.2 Terminologia i esquema

- Backup
- Estratègia de backup (què conservem)
- Procediment de backup (com ho fem)
- Procediment de restauració
- Restauració total/parcial
- Backup incremental/total
- Calendari de backup (quan ho fem)
- Eina de backup
- Backup en calent/en fred
- Dispositiu de backup/suport dels backups (on guardem la informació)
- Armari de backups (on guardem els backups)
- Protocol de crisi (que fem davant un desastre/com ens entrenem)

8.3 Els backups són perillosos

Els backups són una porta d'accés al teu computador amb que no s'hi compta. Computador segur + backup insegur => computador insegur [lleï de la baula més feble].

Endurir la seguretat dels backups sempre comporta més dificultats en cas d'haver de fer una recuperació. Per exemple, si estan encriptats, cal tenir control de les claus d'encriptació que, molt probablement s'hauran perdut en el computador original.

8.4 Els backups són cars

Tenen cost de hardware, de temps de computació [finestra de backup], de personal qualificat i entrenat, d'eines, d'entrenament per poder sobrevenir una crisi.

8.5 Cas simple 1

Suposem un portàtil (màquina pot ser robada fàcilment) que fa de computador personal.

Condicions:

- Suposem backup sobre un disc connectable via USB (del tipus que sigui). Suposem
- Suposem que només el volem per si es perd el portàtil o hi ha una avaria destructiva i es perd.

8.6 Disseny de l'estratègia

- Nomes backups totals
- Només dels fitxers d'usuari, la configuració del sistema i quins paquets instal·lats (per estalviar espai). Això vol dir /etc i /home a més de la llista de paquets.
- Backup en fred: si hi ha processos en calent no podrem garantir la correctesa del backup.
- Sense encriptar (això fa el backup vulnerable)

8.7 Calendari

- Setmanal

8.8 Eina

- ‘duplicity’ (basada en rsync)

8.9 Procediment de backup

1. Desar la llista de paquets instal·lats en un directori que formi part dels directoris dels que es fa backup. Per exemple /etc, Noti's que també es desen les claus que identifiquen els repositoris fiables.

```
$ sudo dpkg --get-selections > /etc/Package.list
$ sudo apt-key exportall > /etc/Repo.keys
```

2. Fer un full-backup cap el disc USB

```
$ sudo duplicity --no-encryption --include /etc --include /home --exclude '**' /
```

8.10 Procediment de recuperació (a entrenar!!)

1. Reinstal·lar mateixa versió de sistema operatiu neteja.
2. Instal·lar duplicity

```
$ sudo apt install duplicity
```

3. Descarregar les claus i llistes de paquets del backup

```
$ sudo duplicity file:///media/XXX /etc/Package.list
$ sudo duplicity file:///media/XXX /etc/Repo.keys
$ sudo duplicity file:///media/XXX /etc/apt/sources.list
$ sudo apt-key add ~/Repo.keys
$ sudo apt update
$ sudo dpkg --clear-selections
$ sudo dpkg --set-selections < /etc/Package.list
$ sudo apt-get dselect-upgrade
```

Compte en tenir el procediment de restore en un script: es pot perdre per culpa del mateix incident i haver de recuperar a ma. Sempre cal tenir clar com es recupera a ma.