

# Gestió de software



# Paquets

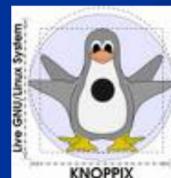
Un paquet de programari és una sèrie de fitxers que es distribueixen de forma conjunta. La raó principal per agrupar aquest fitxers sol ser que el funcionament de cadascun complementa o requereix d'uns altres.

## ■ Tipus de paquets de programari:

- **.rpm (RedHat)**: format dels paquets en distribucions Linux de la familiar Red Hat.
- **.deb (Debian)**: format dels paquets en distribucions Linux de la familiar Debian.
- **.slp (Stampede)**: format de paquets Stampede
- **.tgz (Slackware)**: format de paquet utilitzat a la distribució Linux Slackware
- **.pkg (Solaris)**: format de paquet en sistemes operatius Unix Solaris.

# Distribucions Linux

- Tot i que evidentment hi ha diferències entre distribucions, comparteixen unes característiques bàsiques que faciliten la interoperabilitat.
- Distribucions



# Paquets conceptes

- ◆ **Paquet binari:** són paquets que contenen aplicacions binaries sense el seu codi font. El paquet inclou els fitxers executables/binaris (a les carpetes estàndards segons FHS: /bin/, /usr/bin, /sbin, etc.)
- ◆ **Paquets font:** paquets que inclouen codi font. Normalment cada paquet font té un paquet equivalent binari.
- ◆ **Dependència:** Es parla de dependència quan un paquet requereix que s'instal·li prèviament una altre paquet o paquets, els quals són anomenats dependències.

# Paquets conceptes

- ◆ **Gestor de paquets:** Són les aplicacions que ens permeten gestionar paquets. Són eines que faciliten les tasques més habituals relacionades amb la gestió de paquets (instal·lació, cerques, eliminacions, etc.)
- ◆ **Dipòsit:** un dipòsit (de l'anglès "repository") és un arxiu on s'emmagatzemen paquets. Normalment es troba a Internet. Els dipòsits estan preparats per a ser distribuïts habitualment servint-se d'una xarxa informàtica com Internet, però també els podem trobar en servidors de xarxa LAN (Intranet) o en un mitjà físic com un disc compacte (CD) o DVD.

# Paquets conceptes

- ◆ Paquet virtual (**tonto o dummy**): es tracta d'un paquet buit de continguts amb un nom genèric que proveeix altres paquets mitjançant dependències. (ex. News-reader). Poden haver-hi múltiples aplicacions possibles que es poden configurar amb update-alternatives
- ◆ **Tasques:** Les tasques són agrupacions de paquets que instal·lats de forma conjunta proveeixin una funcionalitat comuna. Per exemple hi ha tasques per a instal·lar un sistema LAMP o entorns d'escriptori com Gnome o KDE.

# Sistema de paquets Debian.

## Debian Packages (extensió .deb)

- ◆ Molt conegut per les seves estrictes polítiques de qualitat abans de alliberar noves versions de paquets.
- ◆ Faciliten l'aplicació d'actualitzacions sense necessitat de reiniciar la màquina.
- ◆ Permet configurar l'aplicació en el moment de la instal·lació (debconf)
- ◆ Permet tasques prèvies a l'eliminació, instal·lació i/o configuració d'un paquet.
- ◆ Són utilitzats ens totes les distribucions de la família Debian
- ◆ El programa principal (canònic) utilitzat per gestionar aquest tipus de fitxers és **dpkg**, tot i que sovint dpkg es utilitzat mitjançant el front-end **apt**.

# dpkg

- Debian package (dpkg)
  - ◆ Aplicació encarregada de gestionar els paquets Debian.
  - ◆ Eina de baix nivell. Normalment com a usuaris utilitzarem altres eines a nivell més alt (com apt-get o aptitude)

```
$ dpkg [options] [action] [package-files|package-name]
```

```
$ whereis dpkg
dpkg: /usr/bin/dpkg /etc/dpkg /usr/lib/dpkg
/usr/share/dpkg /usr/share/man/man1/dpkg.1.gz
$ dpkg -S /usr/bin/dpkg
dpkg: /usr/bin/dpkg
```

# dpkg

## • Accions

```
$ man dpkg-query
```

### ◆ Només consulta (dpkg-query-actions)

- **dpkg -l | --list [<pattern> ...]** : Llista els paquets que compleixen amb un determinat patró.
- **dpkg -s | --status <package> ...** : mostra l'estatus actual d'un paquet i la metainformació del paquet.
- **dpkg -L | --listfiles <package>** : Mostra la llista de fitxers instal·lats
- **dpkg -S | --search <pattern> ...** : Busca un fitxer concret dins dels paquets instal·lats
- **dpkg -p | --print-avail <package> ...** : Mostra detalls sobre un paquet instal·lat.
- **dpkg -I | --info <package>**: Mostra informació sobre un paquet desinstal·lat.

# dpkg

- ◆ Accions que modifiquen el sistema. Cal ser superusuari:
  - **dpkg -i | --install <.deb file name> ...** : Instala un paquet a partir d'un fitxer .deb
  - **dpkg --configure <package> ...** : Torna a configurar un paquet ja instal·la. Executa l'script de post-instal·lació.
  - **dpkg -r | --remove <package> ...** : Elimina un paquet però deixa els fitxers de configuració de sistema.
  - **dpkg -P | --purge <package> ...** : Elimina un paquet incloent els fitxers de configuració.
  - **dpkg -C | --audit** : busca fitxers instal·lats parcialment al sistema i suggereix que fer amb ells.

# dpkg

## • Opcions

- ◆ **--root=<directory>** : permet indicar un altre sistema arrel on efectuar les operacions relacionades amb dpkg (chroot)
- ◆ **-B | --auto-deconfigure**: Es combina amb l'acció remove (-r). Desactiva els paquets que depenen del paquet a esborrar. Instal·la encara que es pugui trencar algun altre paquet.
- ◆ **--force-things**: Opcions force (descarta problemes).
- ◆ **--ignore-dependencies=<package>, ...** : ignora les dependències
- ◆ **--no-act**: comprova dependències i altres accions però no instal·lis o eliminis el paquet. Per veure que passaria però sense fer canvis.
- ◆ **-R | --recursive**. utilitzat amb wildcards (comodins).
- ◆ **-G**: no instal·lis el paquet si hi ha una versió més nova ja instal·lada
- ◆ **-E**: no instal·lis el paquet si ja està instal·lada la mateixa versió

# dpkg -l

- Mostra la informació dels paquets del sistema

```
$ dpkg -l | more
Desired=Unknown/Install/Remove/Purge/Hold
| 
Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig
-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Nom          Versió      Descripció
=====  
ii  acl          2.2.47-2    Access control list
utilities  
ii  acpi-support 0.129       scripts for
ii  acpid         1.0.6-9ubuntu8  scripts for
ii  acpid         1.0.6-9ubuntu8  utilities for using
ACPI power management+  ACPI power management+  Utilities for using
ACPI power management+  ACPI power management+
```

- La primera columna té 3 parts
  - Selection status
  - Status
  - Error status (només apareix en cas que hi hagi algun error)

# dpkg -l

## • Selection status

- ◆ **install (i)**: El paquet esta seleccionat per ser instal·lat.
- ◆ **deinstall(r)**: Els paquet esta seleccionat per ser desinstal·lat
- ◆ **purge (p)**: el paquet esta instal·lat per ser purgat  
(desinstal·la i esborra els fitxers de configuració)
- ◆ **Desconegut (u)**: No es coneix l'estat de selecció

## • Estatus

- ◆ **Installed (i)**: el paquet s'ha desempaquetat i instal·lat correctament
- ◆ **half-installed(h)**: la instal·lació del paquet s'ha començat però no s'ha acabat per algun motiu/error.
- ◆ **Not-installed (n)**: el paquet no esta instal·lat al sistema

# dpkg

- ◆ Package States
  - ◆ **triggers-awaited(W)**: pendent d'un trigger.
  - ◆ **triggers-pending(t)**: El paquet ha estat triggered.
  - ◆ **unpacked(u)**: el paquet està desempaquetat però no configurat.
  - ◆ **half-configured(F)**: el paquet s'ha desempaquetat i la configuració s'ha iniciat però no s'ha acabat per algun motiu/error.
  - ◆ **config-files(c)**: només queden els fitxers de configuració

# dpkg

- ◆ Error status
  - ◆ **hold (h)**: Els paquets marcats amb hold no són controlats per dpkg (excepte que s'utilitzi --force-hold).
  - ◆ **reinst-require (r)**: paquet trencat que necessita ser reinstal·lat
  - ◆ **None**: no apareix cap caràcter
  - ◆ **Both problems (x)**: els dos problemes anteriors

# dpkg

## ◆ Procés d'instal·lació

- ◆ 1. S'estreu el fitxer de control del nou paquet.
- ◆ 2. Si ja hi ha una versió del paquet a instal·lar al sistema (actualització) s'executa l'script **prerm** si existeix.
- ◆ 3. S'executa l'script **preinst** si el paquet proveeix d'aquest script (si existeix).
- ◆ 4. Es desempaqua el nou paquet, al mateix temps és fa una copia dels antics fitxers (permets restaurar en cas d'error)
- ◆ 5. Si ja hi ha una versió del paquet a instal·lar en el sistema (actualització) s'executa l'script **postrm** si existeix.
- ◆ 6. Es configura el paquet (--configure). La configuració del paquet suposa:
  - 6.1. Desempaquetar els fitxers de configuració i al mateix temps fer una copia de seguretat dels antics
  - 6.2. Executar l'script **postinst** si existeix.

# dpkg

- ◆ Procés d'eliminació d'un paquet
  - ◆ 1. S'executa l'script **prerm**.
  - ◆ 2. S'eliminen els fitxers instal·lats (si a més hi ha la opció **--purge** s'eliminen els fitxers de configuració també)
  - ◆ 3. S'executa l'script **postrm**.

# dpkg

## Exemples

**Instal·lació manual:**

```
$ sudo dpkg -i webmin_1.480_all.deb
```

**Desinstal·lació manual:**

```
$ sudo apt-get install hello  
$ sudo dpkg -r hello  
$ sudo dpkg -P hello
```

**Quants paquets hi ha instal·lats al sistema?**

```
$ dpkg -l | wc -l  
1467
```

**Patrons:**

```
$ dpkg -l 'gnome*'
```

**Comprovar si un paquet està instal·lat:**

```
$ dpkg -l | grep nom_del_paquet
```

**Fitxers instal·lats:**

```
$ dpkg -L net-tools
```

**Fitxers binaris:**

```
$ dpkg -L net-tools | grep bin
```

# dpkg

- Obtenir la llista de paquets instal·lats per mida

```
$ dpkg-query -W --showformat='${Installed-Size} ${Package}\n' | sort -nr | more
99400 openoffice.org-core
76696 tetex-base
65824 mysql-server-5.0
65084 sun-java5-bin
64048 linux-image-2.6.17-10-386
60844 linux-image-2.6.15-27-386
60604 linux-image-2.6.15-23-386
58880 linux-headers-2.6.17-10
54740 linux-headers-2.6.15-27
42152 tetex-extra
41520 foomatic-db-gutenprint
39396 openoffice.org-common
38204 evolution
```

# dpkg

- Guardar la llista de paquets instal·lats i restaurar-la posteriorment:

```
$ dpkg --get-selections | grep -v deinstall > ubuntu-files
```

**Per reinstalar:**

```
$ sudo apt-get update
```

```
$ sudo apt-get dist-upgrade
```

```
$ dpkg --set-selections < ubuntu-files
```

# Relacions entre paquets

- ◆ El paquet A depèn del Paquet B: Implica que és absolutament necessari que el paquet B estigui instal·lat abans de poder instal·lar A. També pot ser que A depengui d'una versió específica de B.
- ◆ El paquet A recomana B: B no és necessari però el desenvolupador de A creu que la majoria d'usuaris d'A trobaran útil B.
- ◆ El paquet A suggereix B: El paquet A suggereix el B per què B conté funcionalitats relacionades amb A (sovint milloren la seva funcionalitat).

# Relacions entre paquets

- ◆ **El paquet A entra en conflicte amb el paquet B:** El paquet A no pot operar al sistema si existeix B. Sovint per què tots dos proporcionen el mateix fitxer.
- ◆ **El paquet A reemplaça B:** Al instal·lar A s'esborren els fitxers de B i en alguns casos es sobreescrivien.
- ◆ **El paquet A proveeix de B:** Quan tots els fitxers i funcionalitats de B han estar incorporades a A.

# apt

- ◆ apt (Advanced Packaging Tool)
  - ◆ Utilitat per a la gestió de paquets en sistemes Debian GNU/Linux o derivats.
  - ◆ apt simplifica molt la feina de l'administració de paquets en sistemes UNIX, per què automatitza el procés d'obtenció, instal·lació i configuració de programari.

# apt

- ◆ Apt és un front-end de dpkg.
- ◆ Actualment ja hi ha versions modificades d'apt per treballar amb paquets rpm.
- ◆ A diferència de dpkg, apt instal·la les dependències.
- ◆ No existeix un programa apt en si mateix, sinó que APT és una biblioteca de funcions C++ que s'utilitza per diversos programes de Línia de comandes per a distribuir paquets.

# apt

- ◆ Apt és un conjunt d'eines proporcionades pel paquet apt:

- ◆ Ordres:

```
$ dpkg -L apt | grep bin
/usr/bin
/usr/bin/apt-cache
/usr/bin/apt-cdrom
/usr/bin/apt-config
/usr/bin/apt-get
/usr/bin/apt-key
/usr/bin/apt-mark
```

- ◆ Fitxers de configuració:

```
$ dpkg -L apt | grep etc
/etc
/etc/apt
/etc/apt/apt.conf.d
/etc/apt/apt.conf.d/01autoremove
/etc/apt/apt.conf.d/01ubuntu
/etc/apt/preferences.d
/etc/apt/sources.list.d
/etc/logrotate.d
/etc/logrotate.d/apt
/etc/cron.daily
/etc/cron.daily/apt
```

# apt-get

- ◆ Sincronitzar la base de dades de paquets
  - ◆ Opció update: `$ sudo apt-get update`
  - ◆ La base de dades de paquets (package cache) es local i no es sincronitza automàticament amb els dipòsits (repository). La base de dades es pot gestionar amb apt-cache.
  - ◆ Cada cop que fem un canvi als repositoris, o volem comprovar si tenim les últimes versions dels paquets als repositoris hem d'executar abans un **update**.
  - ◆ S'ha d'executar regularment i sinó tenim paquets com l'**update-manager** que comprova periòdicament si tenim actualització

# apt-get

- ◆ Instal·lar un paquet

```
$ sudo apt-get install pingus
```

- ◆ Es pot simular la instal·lació:

```
$ sudo apt-get --simulate install  
supertux  
supertux
```

- ◆ Altres opcions:

- **-d**: Només descarrega el paquet.
- **-f**: Força la instal·lació encara que falli el test d'integritat.
- **-y**: Respon si a totes les preguntes.

# apt-get

- ♦ Informació durant la instal·lació:

```
$ sudo apt-get install supertux
```

S'està llegint la llista de paquets... Fet

S'està construint l'arbre de dependències

S'està llegint la informació de l'estat... Fet

S'instal·laran els següents paquets **extres**:

  libphysfs1 supertux-data

**S'instal·laran els paquets NOUS següents:**

  libphysfs1 supertux supertux-data

0 actualitzats, 3 nous a instal·lar, 0 a suprimir i 12 no actualitzats.

1 no instal·lats o suprimits completament.

Es necessita obtenir 45,0MB d'arxius.

Després d'aquesta operació s'empraran 63,7MB d'espai en disc addicional.

Voleu continuar [S/n]?

- ♦ **Atenció:** Una instal·lació pot esborrar altres paquets que no siguin compatibles amb el que esteu instal·lant!

# apt-get

- Eliminar un paquet
  - ◆ No elimina els fitxers de configuració

```
$ sudo apt-get install nmap  
$ sudo apt-get remove nmap
```

- ◆ Eliminar completament (purge)

```
$ sudo apt-get remove --purge nmap
```

- ◆ Atenció: al esborrar un paquet també es poden eliminar paquets els quals depenien del paquet esborrat
- ◆ De vegades s'esborren paquets virtuals que poden semblar perillosos de desinstal·lar però no ho son pas

```
$ sudo apt-get remove gimp
```

# apt-get

- autoremove
- Borra paquets del sistema que ja no s'utilitzen:

- Paquets marcats com instal·lats automàticament i que ja no s'utilitzen

```
$ sudo apt-get autoremove
```

```
S'està llegint la llista de paquets... Fet
S'està construint l'arbre de dependències
Reading state information... Fet
The following packages were automatically installed and are no longer required:
libzip-0-12 memcached php4-common
S'ELIMINARAN els següents paquets:
libzip-0-12 memcached php4-common
0 actualitzats, 0 nous a instal·lar, 3 a eliminar i 2 no actualitzats.
Es necessita obtenir 0B d'arxius.
Després de desempaquetar s'alliberaran 664kB d'espai en disc.
Voleu continuar [S/n]? s
(S'està llegint la base de dades ... hi ha 159735 fitxers i directoris
instal·lats actualment.)
S'està desinstal·lant libzip-0-12 ...
S'està desinstal·lant memcached ...
Stopping memcached: memcached.
S'està desinstal·lant php4-common ...
```

# apt-get

- ♦ upgrade

- ◆ Actualitza els paquets instal·lats al sistema dels quals es disposa d'una nova versió al repositori:

```
$ sudo apt-get upgrade  
$ sudo apt-get --simulate upgrade
```

- ♦ dist-upgrade

- ◆ Utilitzat per actualitzar paquets a una nova versió del sistema (canvis de versió a la distribució)
  - ◆ Instal·la els paquets mantinguts

# apt-get

- Esborrar paquets descarregats:
  - ◆ Opció clean

```
$ df -h
S. fitxers          Tamany En ús Lliure
%Ús Muntat en
/dev/sda2           153G   80G   66G
55% /
...
$ sudo apt-get clean
Password:
$ df -h
S. fitxers          Tamany En ús Lliure
%Ús Muntat en
/dev/sda2           153G   79G   67G
55% /
.....
```

- ◆ Els paquets es troben a **/var/cache/apt/archives**

```
/var/cache/apt/archives/amor_4%3a4.3.2-
0ubuntu1_i386.deb
0ubuntu1_i386.deb
```

# apt-get

## • Opció source

- Descarrega paquets de codi font
- Exemple de com descarregar un codi font, compilar-lo i instal·lar-lo:

```
$ sudo apt-get build-dep coreutils  
$ sudo apt-get -b source coreutils  
$ sudo ln -s ~/sources/coreutils-7.4/src  
/usr/src/coreutils-7.4
```

```
$ whereis ls  
ls: /usr/src/coreutils-7.4/ls.c  
/usr/src/coreutils-7.4/ls.o...  
/usr/src/coreutils-7.4/ls.so...  
/usr/src/coreutils-7.4/ls
```

- L'opció **build-dep** instal·la les dependències necessàries per compilar el codi font

# Resol·lució de problemes

- Actualització aturada abruptament durant la instal·lació de paquets

```
# apt-get -f install  
# dpkg --configure -a
```

- Mai poden haver-hi més d'una aplicació de gestió de paquets obertes al mateix temps
  - Si una aplicació de gestió de paquets es penja pot deixar bloquejat l'accés a la base de dades de

```
E: No s'ha pogut blocar /var/lib/dpkg/lock - open (11: Resource  
temporarily unavailable)  
E: No s'ha pogut blocar el directori d'administració  
(/var/lib/dpkg/). Hi ha algun altre procés que l'estigui utilitzant?  
$ sudo rm /var/lib/dpkg/lock
```

# Resolució de problemes

- ◆ No es pot eliminar un paquet. Error en algun del fitxers d'script (p. ex. Postrm)

- ◆ Els fitxers d'script es troben a:

```
/var/lib/dpkg/info/
```

- ◆ Per exemple els postrm

```
$ ls /var/lib/dpkg/info/*.postrm
```

- ◆ Es pot eliminar l'script o posar un return 0 al principi del fitxer

# apt-cache

- apt-cache

- És l'eina que permet manipular la memòria cau (cache) dels paquets Debian. L'ordre només serveix per proporcionar informació sobre la base de dades de paquets Debian (també coneguda com package cache).

- Buscar paquets:

```
$ apt-cache search
```

word

- Dependències:

```
$ apt-cache depends
```

nom del paquet

- Informació d'un paquet:

```
$ apt-cache show
```

nom del paquet

# apt-cache

- ◆ Informació detallada 

```
$ apt-cache showpkg package
```
- ◆ Estadístiques de la base de dades de paquets

```
$ sudo apt-cache stats
```

- ◆ Noms dels paquets:  

```
$ apt-cache pkgnames
```

**Número de paquets:**  

```
$ apt-cache pkgnames |
```

$$\frac{\text{wc} - 1}{\text{wc}}$$
- ◆ Dependències no satisfetes:

```
$ sudo apt-cache unmet
```

- ◆ Informació detallada d'un paquet

```
$ apt-cache showpkg paquet
```

# Configuració de dipòsits (repositoris)

## ◆ Afegir un repositori amb clau

Afegir la següent línia a l'arxiu /etc/apt/sources.list:

```
deb http://www.array.org/ubuntu intrepid eeepc
```

Descarregar la clau:

```
$ wget http://www.array.org/ubuntu/array-apt-key.asc
```

Instal·lar la clau:

```
$ sudo apt-key add array-apt-key.asc
```

També ho podem fer amb una sola ordre:

```
$ wget http://www.array.org/ubuntu/array-apt-key.asc -O - |
\ sudo apt-key add -
```

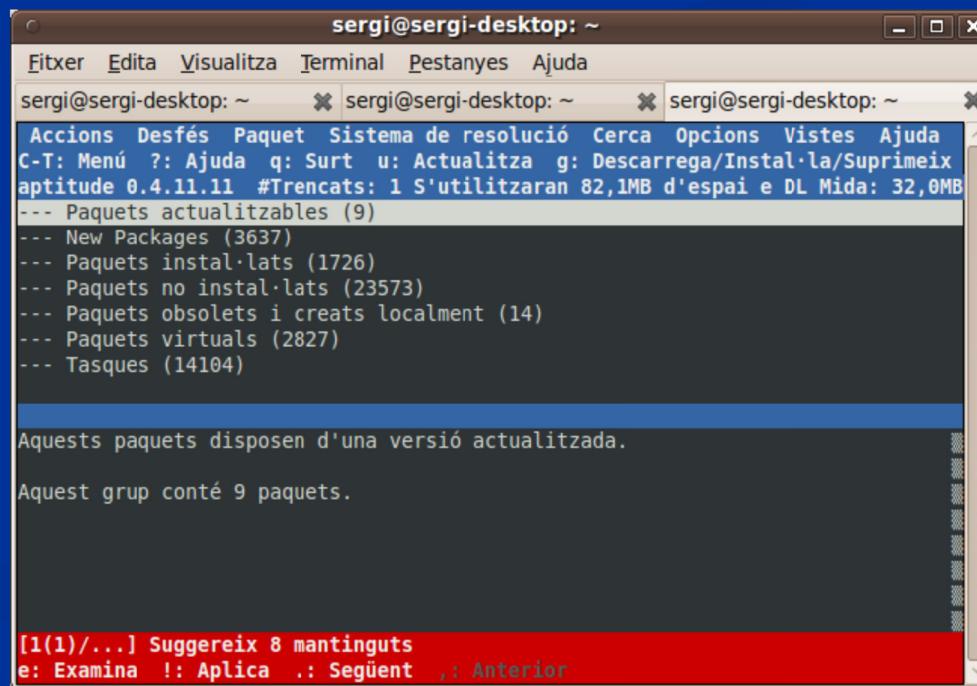
I finalment cal actualitzar gestor de paquets:

```
$ sudo apt-get update
```

# aptitude

- Interfície gràfica
  - ◆ Accepta les mateixes opcions que apt-get.
  - ◆ L'única diferència es que esborra les dependències als esborrar un paquet (no necessita l'ordre clean)

```
$ sudo aptitude
```



# aptitude

- Tecles
  - ◆ **u**: \$ sudo aptitude update
  - ◆ **U**: \$ sudo aptitude upgrade
  - ◆ **q**: quit
  - ◆ **+**: Marcar per instal·lar
  - ◆ **-**: Marcar per desinstal·lar
  - ◆ **gg**: El primer g mostra les tasques pendents. Els segon g aplica.
  - ◆ **?**: Ajuda
  - ◆ **/nom**: Cercar un paquet per nom

# aptitude

- ◆ Instal·lar un paquet
  - ◆ Localitzem el paquet en la categoria «Paquets no instal·lats», usant les tecles del cursor del teclat i la tecla INTRO.
  - ◆ Seleccionem el paquet que desitgem instal·lar i premem la tecla + el paquet es posarà en verd.
  - ◆ Premem la tecla g i es presentarà un resum de les accions que es van a realitzar.
  - ◆ Premem g una altra vegada, i començarà la descàrrega i posterior instal·lació del paquet.

# Conversió de paquets

- ◆ Alien:
  - ◆ Convertim el paquet "package.rpm" a format .deb en el paquet "package.deb"
    - **\$ alien --to-deb package.rpm**
  - ◆ Convertim el paquet "package.deb" a format rpm en el paquet "package.rpm"
    - **\$ alien --to-rpm package.deb**
  - ◆ Podem fer conversions múltiples:
    - **\$ alien --to-deb --to-rpm --to-tgz --to-slp foo.deb bar.rpm baz.tgz**

# Sistema de fitxers Linux

- ◆ Sistemes de fitxers Unix/Linux
  - ◆ Tots els fitxers de tots els dispositius es troben en una sola jerarquia anomenada **Sistema Virtual de fitxers**.
  - ◆ L'arrel d'aquesta jerarquia única s'anomena **root (arrel)** i és representat amb el caràcter /
  - ◆ A Linux/Unix **tot són fitxers** (fitxers, carpetes, dispositius de maquinari, enllaços, sockets, recursos remots...)

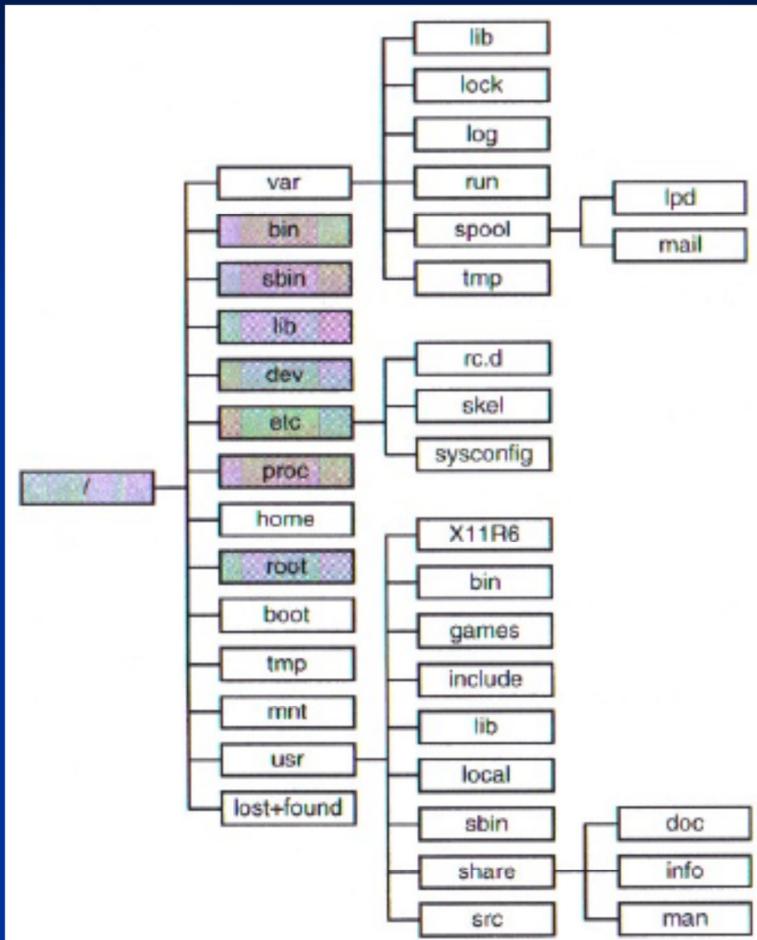
# Sistema de fitxers Linux

- ◆ Una de les particularitats de Linux és presentar els dispositius connectats al sistema en forma d'arxius. Per exemple, la carpeta **/dev** conté els fitxers que representen els dispositius de maquinari del sistema (**/dev/fd0** és el disquet i **/dev/cdrom** és el CD-ROM).
- ◆ Els fitxers no han de ser d'un disc dur concret i ni tan sols han de ser de la màquina local (es pot tenir l'arrel en una màquina remota)

# Sistema de fitxers Linux

- ◆ Per accedir a un dispositiu primer s'ha d'informar al Sistema Operatiu del camí dins de la jerarquia principal des d'on es podrà accedir als fitxers del dispositiu
  - Aquest procés s'anomena **muntatge de dispositius**
  - El directori des de el qual es pot accedir al dispositiu s'anomena **Punt de Muntatge**.
  - Generalment només el super usuari (root) pot muntar dispositius o indicar quins dispositius poden ser muntats pels usuaris.

# FHS



- Filesystem Hierarchy Standard
- El sistema de fitxers de Linux segueix un estàndard anomenat FHS
  - La majoria de distribucions Linux i els sistemes Unix segueixen el mateix estàndard.

# FHS

## Objectius

- ◆ Permetre al programari predir la localització dels fitxers i directoris a instal·lar
- ◆ Permetre als usuaris predir on el programari instal·larà els fitxers i els directoris.
  - Especifica el mínim de fitxers i directoris necessaris
  - Especifica quin és l'objectiu de cada àrea del sistema
  - Enumera les excepcions
  - Enumera els conflictes històrics
- ◆ Les aplicacions, distribucions i sistemes operatius que segueixen l'estàndard són anomenades FHS complaint

# FHS

- 4 tipus de fitxers
  - ◆ **shareable | unshareable**: són els fitxers que es poden o no es poden compartir entre màquines diferents.
  - ◆ **variable | static**: fitxers que no es modifiquen amb el temps (excepte noves versions) o fitxers que tenen canvis de forma continuada.

	shareable	unshareable
static	/usr	/etc
	/opt	/boot
variable	/var/mail	/var/run
	/var/spool/news	/var/lock

# FHS

- ◆ Arriba a especificar quines són les ordres imprescindibles, (carpeta /bin)
  - ◆ cat, chgrp, chmod, chown, cp, date, dd, df, dmesg, echo, false, hostname, kill, ln, login, ls, mkdir, mknod, more, mount, mv, ps, pwd, rm, rmdir, sed, sh, stty, su, sync, true, umount, uname
  - ◆ I també els fitxers de configuració de la carpeta /etc
- ◆ 2 nivells
  - ◆ Sistema (nivell primari)
  - ◆ Usuari: /usr (nivell secundari)

# Jerarquia de Fitxers

- ◆ Executables (binaris)
  - ◆ **/bin:** executables bàsics pel funcionament del sistema. P. ex. comandes com ls.
  - ◆ **/sbin:** “super” binaris. Executables del superusuari.
  - ◆ **/usr/bin i /usr/sbin:** executables secundaris
- ◆ Llibreries
  - ◆ **/lib:** llibreries principals dels executables principals (carpetes **/bin i /sbin**)
  - ◆ **/usr/lib:** llibreries secundaries de la resta d'executables.

# Jerarquia de Fitxers

- ◆ Homes (carpetes d'usuari)
  - ◆ **/home:** Conté els directoris personals dels usuaris de la màquina. P. ex. els usuaris juan, ana i pep tindran les següents homes:
    - /home/
    - juan
    - /home/ana
    - /home/pep
- ◆ Fitxers de configuració
  - ◆ **/etc:** directori (“etcètera”) conte els fitxers de configuració i scripts d’arrancada del sistema.
    - **/etc/rc.d** conté els scripts d’arrancada i control de serveis.
    - **/etc/skel** (director “esquelet”) conté els arxius que es copiaran al directori de l’usuari en crear un nou compte.
    - **/etc/X11:** conte la configuració del sistema gràfic.

# Jerarquia de Fitxers

- ◆ /usr
  - ◆ Jerarquia secundaria on s'emmagatzemen les dades de només lectura compartides entre usuaris. És la carpeta més voluminosa del sistema
  - ◆ **bin/sbin:** executables.
  - ◆ **/usr/include:** fitxers include de c (fitxers \*.h).
  - ◆ **/usr/lib:** llibreries secundaries.
  - ◆ **/usr/games:** conté els executables dels jocs.
  - ◆ **/usr/share:** dades compartides (icones, pixmaps, doc, i18n, manuals unix, etc.)
  - ◆ **/usr/src:** codi font de programes.
  - ◆ **/usr/X11R6:** El sistema gràfic X11 (release 6)

# Jerarquia de Fitxers

## ◆ Dades “variables”

- ◆ **/var:** Es troben tot els arxius de dades “variables” en el temps: cues d’espera de la impressora, bústies dels usuaris, registres (logs) del sistema, bases de dades.
- ◆ Part del disc amb accés de lectura i escriptura continu. Sovint és munta en una partició a part.
  - **/var/games:** Dades variables dels jocs
  - **/var/lib/{nom\_aplicació}:** Bases de dades de les aplicacions
  - **/var/log:** Registres del sistema.
  - **/var/mail:** Missatgeria local Unix.
  - **/var/tmp:** Un altre espai temporal.
  - **/var/run:** indicadors de les aplicacions que s'estan executant.
  - **/var/lock:** indicadors de bloqueig de les aplicacions.
  - **/var/spool:** Cues del sistema (correu electrònic, impressores, etc)

# Jerarquia de Fitxers

- ◆ /mnt
  - ◆ Punts de muntatge temporals. Per exemple:
    - /mnt/windows: Accés a la partió de Windows
- ◆ /media
  - ◆ Punts de muntatge per als dispositius removibles com CD/DVD-ROMS, llapis USB. etc:
    - /media/cdrom
    - /media/usbdisk
- ◆ /dev
  - ◆ Conté arxius com /dev/fd0 i /dev/cdrom per representar la disquetera i la unitat de CD.
  - ◆ /dev/null és com una paperera.

# Jerarquia de Fitxers

- ◆ /root
  - ◆ Home del superusuari
- ◆ /tmp
  - ◆ S'utilitza per emmagatzemar els arxius temporals.
- ◆ Tots els usuaris del sistema poden escriure en aquest directori però l'administrador planifica generalment una neteja automàtica d'aquest directori a intervals regulars (normalment al reiniciar sessió es perden els fitxers)
- ◆ /lost+found
  - ◆ Es crea automàticament a l'arrancada del sistema. L'utilitza la eina de verificació fsck (“File System Check”) per guardar els arxius recuperats després d'un incident en el sistema. Si tot va bé, aquest directori hauria de romandre buit.

# Jerarquia de Fitxers

- ◆ /proc
  - ◆ Sistema d'arxius virtual que representa l'estat del sistema. No ocupa espai en el disc (només existeix en memòria RAM). Cada arxiu que conté dóna accés a informació del sistema com l'ús actual de la memòria, dispositius detectats, enrutament dels paquets de la xarxa, etc.
  - ◆ Per cada procés executat en el sistema, existeix un subdirectorí que el caracteritza en /proc.
- ◆ /boot
  - ◆ Conté el nucli de Linux i altres arxius que s'executen durant l'arrancada del sistema. Es pot trobar en una partició diferent.
- ◆ /opt
  - ◆ Aplicacions proporcionades per tercets

# Buscar

- Buscar fitxers:
  - ◆ **Ordre find:** busca directament als dispositius del sistema
  - ◆ **Ordre locate:** busca en un índex (base de dades).
- Tenim 3 eines que ens poden ajudar a buscar ordres
  - ◆ **which:** busca executables dins del PATH
  - ◆ **whereis:** busca executables, manuals i codi font
  - ◆ **type:** mostra que s'executarà realment al escriure un text a l'interpret d'ordres

# Buscar fitxers. Ordre find

- ◆ Permet buscar fitxers

- ◆ La cerca és fa en el moment d'utilitzar l'ordre (no s'utilitzen índexs com a l'ordre locate)
- ◆ Les cerques sense índexs són més lentes

- ◆ Sintaxi

```
$ find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]
```

- ◆ Les opcions no s'utilitzen gaire (opcions d'optimització i de com treballar amb enllaços simbòlics). El més important són les expressions:

```
$ find path1 [path2]... expression
```

# Buscar fitxers. Ordre find

- ◆ Expressió per defecte: mostrar (-print)
  - ◆ Per defecte mostra tots els fitxers recursivament

```
$ find . = find . -print
```

- ◆ Exemples
  - ◆ Qualsevol tipus de fitxer pel seu nom (tot el sistema)
  - ◆ Només fitxers regulars (no directoris) en el directori de treball)
  - ◆ Només directoris regulars per nom (poder llegir tots els fitxers)

```
$ find / -name nom_fitxer
```

```
$ find / -type f -name nom_fitxer
```

```
$ sudo find / -type d -name nom_carpet
```

# Buscar fitxers. Ordre find

- ♦ Mostrar els directoris de l'arrel

```
$ find / -maxdepth 1
```

```
-type d
```

- ♦ Mostrar els directoris de l'arrel i els seus principals subdirectoris amb informació de ls

```
$ find / -maxdepth 2 -type d -ls
```

- ♦ Buscar fitxers segons els seus permisos

```
$ find . -perm 664
```

- ♦ Buscar fitxers amb permisos SUID OR GUID

```
find . -type f \(\ -perm -04000 -o  
-perm -02000 \)\  
-perm -02000 \)
```

- ♦ Fitxers d'un usuari o grup

```
$ sudo find / -group grup  
$ sudo find / -user usuari
```

- ♦ Executar una ordre a tots els fitxers trobats

```
find . -type f -exec  
file {} \;
```

- ♦ Buscar text dins de múltiples fitxer (millor utilitzar grep)

# Buscar fitxers. Ordre find

- ◆ Canviar recursivament els permisos de només els fitxers

```
find . -type f -exec chmod 644 {} \;
```

- ◆ Fitxers més antics de 30 dies:

```
$ find /linux2/backups/mysql -type f -mtime +30
```

- ◆ Buscar fitxers modificats durant les últimes 24 hores

```
$ find $HOME -mtime 0
```

- ◆ Esborrar fitxers

```
find . -name DEADJOE -delete
```

- ◆ Fitxers que continguin una paraula

```
find . -name *linux*
```

- ◆ Fitxers grans i guardar-los a un fitxer

```
find ~ -size +100M > big.txt
```

# Buscar fitxers. Ordre find

- Buscar els fitxer modificats els últims 10 minuts

```
find ~ -type f -mmin -10 | xargs ls -l
```

- Expressions regulars i ignorar fitxers

- Buscar els fitxers modificats les últimes 24 hores però ignorant les carpetes ocultes (que no els fitxers)

```
$ find ~ \(\ ! -regex '.*/\..*/..*' \) -type f -mtime 0
```

# locate

- Instal·lació

- ◆ Debian: `$ sudo apt-get install mlocate`

- ◆ rpm (Open Suse) Paquet: `findutils-locate-4.4.0-38.27`

- Us

- `$ locate document`

- ◆ Accepta file globbing. Per defecte cerca **\*paraula\***
  - ◆ Case sensitive (opció `-i --ignore-case`)
  - ◆ Suporta expressions regulars amb `--regex`

- **-b, --basename:** només busca al base name (el nom del fitxer sense el path complet). É
    - **-c, --count:** Mostra el número de coincidències en la cerca.
    - **-e, --existing:** s'assegura que el fitxer encara existeix abans de mostrar-lo
    - **-S, --statistics:** Mostra estadístiques.
    - **-q --quiet:** no mostris errors

# locate

- Fitxers
  - Base de dades + cron + ordres (updatedb, locate) + fitxer de configuració

```
$ dpkg -L mlocate  
/.  
/var/lib/mlocate  
/etc  
/etc/updatedb.conf  
/etc/cron.daily  
/etc/cron.daily/mlocate  
/usr/share/man  
...  
/usr/share/doc  
...  
/usr/bin/mlocate  
/usr/bin/updatedb.mlocate
```

Actualització diària a les 6:25 (o la següent engegada després d'aquesta hora )

```
$ cat /etc/cron.daily/mlocate
```

# locate

- ◆ Base de dades

- ◆ /var/lib/mlocate/mlocate.db
- ◆ S'actualitza/crea amb updatedb
- ◆ S'autoexecuta amb cron cada dia (cron.daily)
- ◆ Podem actualitzar a mà amb:
- ◆ Estadístiques:

```
$ sudo updatedb&
```

```
$ locate -S
Base de dades /var/lib/mlocate/mlocate.db:
    65629 directoris
    457738 fitxers
    39433567 bytes en els noms de fitxers
    14517153 bytes usats per a emmagatzemar
    la base de dades
```

# Locate. Fitxer de configuració

- ◆ /etcupdatedb.conf
  - ◆ To prune: podar, retallar
  - ◆ Determina quins fitxers no s'indexen a la base de dades:
    - PRUNENAME: Fitxers o extensions
    - PRUNEWAYS: Carpetes que no s'indexen
    - PRUNEFSS: Sistemes de fitxers (especials, remots...)

```
$ cat /etcupdatedb.conf
PRUNE_BIND_MOUNTS="yes"
# PRUNENAMES=".git .bzr .hg .svn"
PRUNEWAYS="/tmp /var/spool /media"
PRUNEFSS="NFS nfs nfs4 rpc_pipefs afs btrfs_misc proc smbfs autofs iso9660
ncpfs coda devpts ftpfs devfs mfs shfs sysfs cifs lustre_lite tmpfs usbfs udf"
```

# locate

- ◆ Inconvenients

- ◆ **Espai:** La base de dades ocupa un espai de disc. La mida de la base de dades no es gaire gran i en la majoria de sistemes actuals aquest inconvenient no és gaire important
- ◆ **Sincronització:** Cada vegada que hi ha una modificació al sistema de fitxers, hauríem de tornar a executar updatedb per tal de mantenir la base de dades sincronitzada.
- ◆ Es necessiten **permisos de superusuari** per modificar la base de dades.

# whereis

- Localitza l'executable, el codi font i el manual relacionat amb el text proporcionat:
  - ◆ Podem cercar més que ordres. P. ex. crides de sistema

```
$ sudo apt-get install manpages-dev
$ whereis fork
fork: /usr/share/man/man2/fork.2.gz
```

# whereis

- Executable + manual:

```
$ whereis ls  
ls: /bin/ls  
/usr/share/man/man1/ls.1.gz  
/usr/share/man/man1/ls.1.gz
```

- Opcions. Permeten controlar que mostrar i on buscar

- -b | -B: Busca només el binari | Determina on buscar els binaris.
- -m | -M : Busca només el manual. | Determina on buscar el manual
- -s | -S : Busca només el codi font | Determina on buscar el codi font

# which

- ◆ Mostra el path absolut d'una ordre
  - ◆ Només busca els executables a les carpetes indicades a la variable d'entorn PATH
  - ◆ Funciona millor **whereis**
  - ◆ No funciona amb ordres internes

```
$ which ls  
/bin/ls  
$ which cd  
---
```

# type

- ◆ Ordre interna de bash
  - ◆ Ens indica com s'executarà un text concret a la bash

```
$ type cd  
cd is a shell builtin  
  
$ type ls  
ls is aliased to `ls --color=auto',  
      color=auto'
```

- ◆ Ens mostra el PATH complet de l'ordre i es revela l'àlies (si hi ha).
- ◆ Identifica les ordres internes

```
$ type type  
type is a shell builtin
```

# Webgrafia i/o material

## ■ Tutorial

- <http://www.aboutdebian.com/packages.htm>
- [http://www.comptechdoc.org/os/linux/usersguide/linux\\_ugfinding.html](http://www.comptechdoc.org/os/linux/usersguide/linux_ugfinding.html)