

---

# Contingut

---

1. INICIALIZACIÓ DEL SISTEMA
2. INICIALIZACIÓ D'UNA SESSIÓ
3. CONFIGURACIÓ DE LES LOCALES
4. CONFIGURACIÓ DE PAQUETS
5. CONFIGURACIÓ DE LA XARXA
6. LOGGING - rsyslogd -
7. USUARIS I GRUPS

---

# 1. INICIALIZACIÓ DEL SISTEMA

---

Pas 1: Arranc del **Nucli**. La seva imatge sol ser `/boot/vmlinuz`. Per assegurar-vos-en editeu el fitxer `/etc/lilo.conf` (si el modifiqueu heu d'executar `# lilo`).

Pas 2: Execució del procés `init` (procés amb `pid=1`). Passos:

1. Execució dels scripts situats en `/etc/rcS.d/`

2. Execució dels scripts situats en `/etc/rcx.d/`

- La **x** depèn del nivell d'execució (runlevel) del procés `init`. Nivells: (0: halt; 1: single; 2, 3, 4, 5: nivells d'execució normals; 6: reboot).

---

# 1. INICIALIZACIÓ DEL SISTEMA

---

- El nivell d'execució s'especifica en el fitxer `/etc/inittab`:

```
id:2:initdefault:
```

O bé en el fitxer `/etc/init/rc-sysinit.conf`:

```
env DEFAULT_RUNLEVEL=2
```

En els dos casos, el nivell d'execució és el **2**

- Per saber el nivell d'execució actual: `$ runlevel`
- Per canviar el nivell: `$ init 4 // canvia al nivell 4`

# 1. INICIALIZACIÓ DEL SISTEMA

---

Els scripts situats tant en `/etc/rcS.d/` com en `/etc/rcx.d/` són enllaços simbòlics a scripts situats en `/etc/init.d/`. P.e.:

```
/etc/rcS.d/S35mountall.sh -> ../init.d/mountall.sh
```

```
/etc/rc2.d/K45apache -> ../init.d/apache
```

on:

**S,K:** indiquen script a executar al iniciar-se (**S**: start) o finalitzar-se (**K**: Kill) el nivell

**nombre natural:** ordre d'execució (en l'exemple 35, 45)

**nom:** normalment nom de l'script situat en `/etc/init.d`

---

## 2. INICIALIZACIÓ D'UNA SESSIÓ

---

1. Execució del fitxer `/etc/profile`
2. Execució del fitxer `/etc/bash.bashrc` (suposant que bash és l'interpret de comandes per defecte)
3. Execució del fitxer `~/ .profile`, on `~` és el directori HOME de l'usuari en qüestió
4. Normalment, dins del fitxer `./profile` es crida el fitxer abans d'acabar `~/ .bashrc` (suposant que bash és l'interpret de comandes per defecte)
5. Normalment, dins del fitxer `~/ .bashrc`, abans d'acabar, es crida el fitxer `/etc/bash_completion`

---

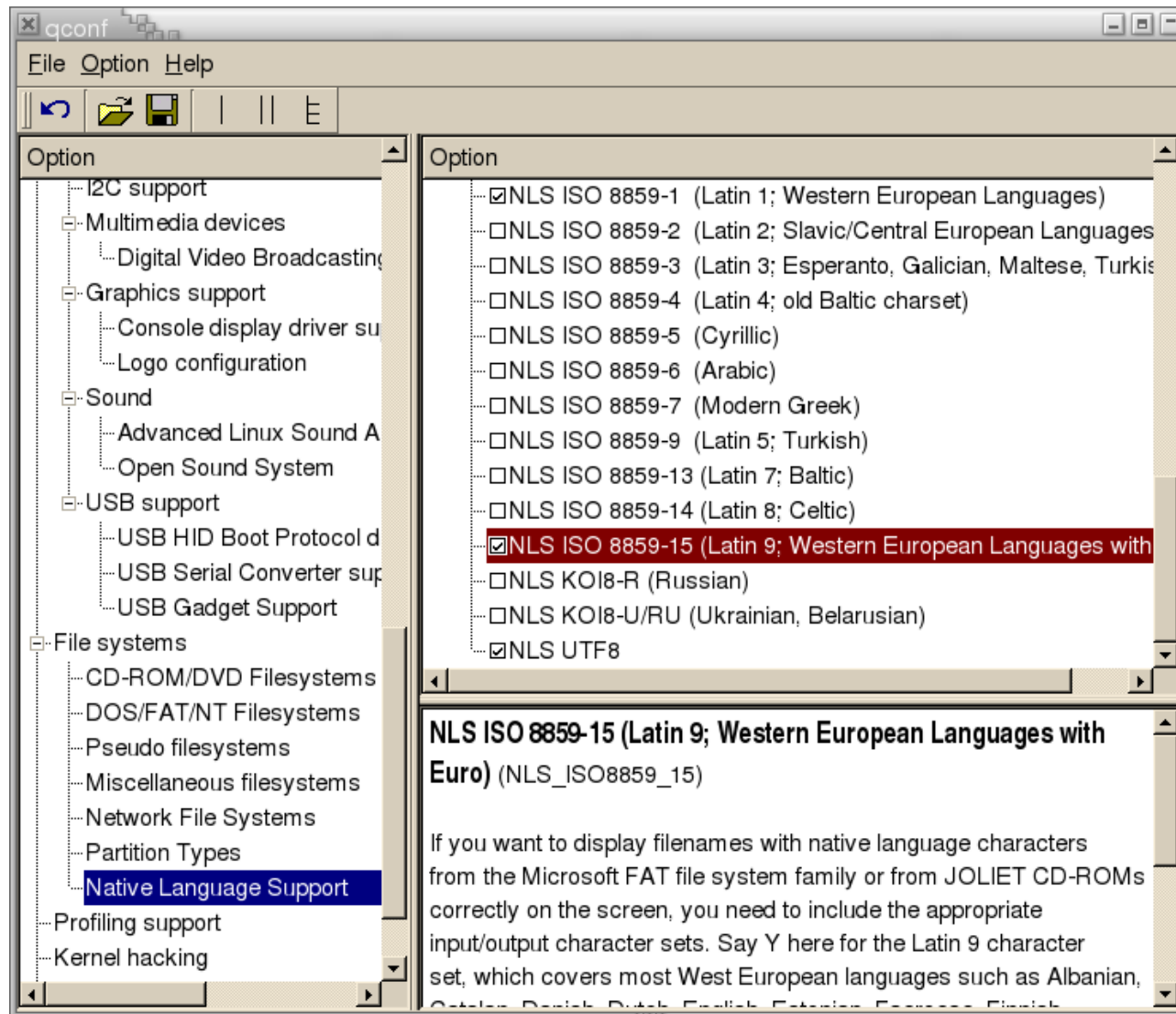
## 3. CONFIGURACIÓ DE LES LOCALES

---

- **LOCALES:** idioma i tipus de codificació dels caràcters. Pel català o el castellà, com tenen caràcters accentuats, hem d'utilitzar tipus de codificacions de 8 bits. Paquets necessaris: `locales` (`localeconf` no cal).
- Tipus de codificacions:
  - C/POSIX: utilitza codificació de caràcters de 7 bits. Solament va bé per l'anglès.
  - ISO-8859-1 (occidental): codificació de caràcters de 8 bits. És necessari tenir suport en el nucli.
  - ISO-8859-15 (occidental): actualització del ISO-8859-1. És l'òptima tant pel **català** com pel **castellà**. És necessari tenir suport en el kernel.

### 3. CONFIGURACIÓ DE LES LOCALES - kernel -

Suport al nucli per iso-8859-1 i iso-8859-15



---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Format arxiu “/etc/apt/sources.list” (adreces dels paquets):

```
deb[-src] URI distribució [secció1] [secció2] ..
```

- deb (paquets binari); deb-src (paquets codi font).
- URI: file, cdrom, ftp, http..
- secció[i]: normalment és un subdirectori.
- Exemples:

```
deb-src http://security.ubuntu.com/ubuntu precise-security main restricted
```

```
deb http://security.ubuntu.com/ubuntu precise-security universe
```



---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Actualitzant llista de paquets: `# apt-get update`
- Actualitzant paquets: `# apt-get upgrade`
- Millor actualització: `# apt-get [-u] dist-upgrade`
- Instal·lant paquets: `# apt-get install xchat [[gnome] [kde]]`
- Eliminant paquets: `# apt-get remove gnome-panel`
- Eliminació completa: `# apt-get --purge remove gnome-panel`

---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Llistant els paquets:

```
# apt-show-version
```

```
.....
```

```
autoconf/testing uptodate 2.59-8
```

- Llistant els paquets que tenen actualització:

```
# apt-show-version -u
```

```
.....
```

```
acroread 5.09-woody0.0 newer than version in archive
```

---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Obtenint noms de paquets:

```
# apt-cache search mplayer
```

```
mplayer - movie player for Unix-like systems
```

```
# apt-cache search movie player for Unix-like systems
```

```
mplayer - movie player for Unix-like systems
```

- Obtenint informació de paquets:

```
# apt-cache show mplayer
```

- Per llistar les dependències d'un paquet:

```
# apt-cache depends mplayer
```

---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Per saber el paquet que conté un cert fitxer:

```
# dpkg -S stdio.h
```

```
gcc-3.0: /usr/lib/gcc-lib/i386-linux/3.0.4/include/stdio.h
```

```
# apt-file search <fitxer>
```

mostra també paquets sense instal·lar que contenen <fitxer>

- Averiguant els paquets que estan instal·lats:

Tots els paquets: 

```
# dpkg -l
```

Hi ha un paquet instal·lat de mozilla? 

```
# dpkg -l | grep mozilla
```

---

## 4. CONFIGURACIÓ DE PAQUETS

---

### Descarregant paquets font (1/2)

S'ha de tenir almenys una entrada del tipus “deb-src” en  
“sources.list”

1. Primer de tot s'han d'instal·lar els paquets que depenen del paquet nou:

```
# apt-get build-dep <paket>
```

Si només es volen llistar:

```
# apt-cache showsrc <paket>
```

2. Descarregant un paquet font (un dels dos):

a) Solament descàrrega del paquet:

```
# apt-get source <paket>
```

descarrega el paquet en el directori en curs.

---

## 4. CONFIGURACIÓ DE PAQUETS

---

### Descarregant paquets font (2/2)

b) Descarregant i alhora compilant un paquet font:

```
# apt-get -b source <paket>
```

a més compila el paquet i deixa el paquet en format .deb.

3. Creant un paquet .deb. En el directori <paket-versió> executar:

```
# dpkg-buildpackage -rfakeroot -uc -b
```

Es podrà instal·lar el paquet normalment amb:

```
# dpkg -i <paket-versió>.deb
```

---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Obtenció dels fitxers d'un paquet:

```
# apt-file list <paket>
```

- Obtenció dels paquets de forma automàtica:

```
# auto-apt run <comanda> // instal·la paquets necessaris per  
<comanda>
```

- Instal·lació fallida. Intenta les comandes següents:

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

- Reconfigurant paquets:

```
# dpkg-reconfigure gdm # dpkg-reconfigure -a
```

- Com obtenir el millor source:

```
# netselect ftp.debian.org ftp.rediris.es  
341 ftp.rediris.es// ftp.rediris.es millor host (341: puntu-  
ació)
```

---

## 4. CONFIGURACIÓ DE PAQUETS

---

- Utilitat `dpkg`:

```
# dpkg -i <paquet> // inserta un nou paquet
```

```
# dpkg -r <paquet> // elimina un paquet
```

```
# dpkg -l // llista tots els paquets installats
```

```
# dpkg -l | grep mplayer // llista tots els paquets que con-  
tenen mplayer
```

- Utilitat `dselect` (pas [3] Instal·la)

+ (-) : seleció d'un paquet a ser instal·lat (desinstal·lat)

↑X : desfà totes les seleccions prèvies

/<paquet> : cerca <paquet>



---

## 5. CONFIGURACIÓ DE LA XARXA

---

1. S'executa `/etc/rc0.d/S35networking` que apunta a `/etc/init.d/networking`. En executar-se (amb l'argument `start`) es crida `ifup` (més concretament “`ifup -a`”, la qual inicialitza tots (opció “`-a`”) els dispositius de xarxa)
2. Per configurar els dispositius de xarxa, la comanda `ifup` té en compte el fitxer de configuració `/etc/network/interfaces`
3. El fitxer `/etc/network/interfaces` informa de com s'ha de configurar la xarxa. Per fer-ho s'utilitza les **paraules clau** següents:
  - (a) **auto**: automàticament (en l'arranc del sistema, quan es fa “`ifup -a`”)
  - (b) **dhcp (static)**: per `dhcp` per obtenir una adreça de forma dinàmica (o amb adreça estàtica)

---

## 5. CONFIGURACIÓ DE LA XARXA - DHCP -

---

Fitxer `/etc/network/interfaces`

**# The loopback interface**

```
auto lo
```

```
iface lo inet loopback
```

**# The network interface**

```
auto eth0
```

```
iface eth0 inet dhcp
```

---

## 5. CONFIGURACIÓ DE LA XARXA - Adreça fixa -

---

Fitxer `/etc/network/interfaces`:

### # The loopback network interface

```
auto lo
```

```
iface lo inet loopback
```

### # The primary network interface

```
auto eth0
```

```
iface eth0 inet static
```

```
    address 192.168.1.90
```

```
    gateway 192.168.1.1
```

```
    netmask 255.255.255.0
```

```
    network 192.168.1.0
```

```
    broadcast 192.168.1.255
```

---

## 5. CONFIGURACIÓ DE LA XARXA

---

### Ordres útils

```
$ hostname
```

```
francesc-VirtualBox
```

```
$ domainname
```

```
(nome)
```

```
$ /sbin/ifconfig
```

```
eth0 Link encap:Ethernet HWaddr 08:00:27:9f:b6:af  
      inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0  
      inet6 addr: fe80::a00:27ff:fe9f:b6af/64 Scope:Link  
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
      RX packets:41057 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:24924 errors:0 dropped:0 overruns:0 carrier:0  
      collisions:0 txqueuelen:1000  
      RX bytes:30677670 (30.6 MB) TX bytes:1829477 (1.8 MB)
```

---

## 5. CONFIGURACIÓ DE LA XARXA

---

- Altres fitxers de configuració:

1. `/etc/hosts` //conté el nom del host i els alias. Per exemple:  
`127.0.0.1 localhost.localdomain localhost`  
`10.50.24.15 eps01.udl.net eps01`
2. `/etc/resolv.conf` //conté el domini i els dns's. Per exemple:  
`domain udl.net`  
`search udl.net`  
`nameserver 10.69.1.1`

---

## 6. LOGGING - rsyslogd -

---

Iniciant syslogd: **rsyslogd**

Fitxer de configuració (conté regles): `/etc/rsyslog.d/50-default.conf`

Format regla: **selector** **action**

**selector** format: **facility.priority**

- **facility** keywords: auth, authpriv, **cron**, daemon, **ftp**, kern, lpr, mail, **mark**, news, security (same as auth), syslog, user, uucp, local0→local7
- **priority** keywords (in ascending order): debug, info, notice, warning, warn (same as warning), err, error (same as err), crit, alert, emerg, panic (same as emerg). warn, error and panic are deprecated.

---

## 6. LOGGING - rsyslogd -

---

Special keywords:

“**\***” : all facilities or priorities.

“**none**” : no priority of the given facility. Used to exclude

“**,**” : multiple facilities (comma separated facilities)

“**;**” : multiple selectors (colon separated selectors)

“**=**” specify only this single priority and not any higher priority

“**!**” ignore all that priorities and any higher priority

“**-**” : omit syncing the file after every logging

“**|**” : pipe. Useful for debugging (to apply program filters)

## 6. LOGGING - rsyslogd -



```
# /etc/syslog.conf. For more information see syslog.conf
# Standard logfiles. Log by facility
auth,authpriv.* /var/log/auth.log
*. *;auth,authpriv.none -/var/log/syslog
#cron.* /var/log/cron.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
lpr.* -/var/log/lpr.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
uucp.* /var/log/uucp.log
# Logging for the mail system. info and higer priorities to var/log/mail.info
mail.info -/var/log/mail.info
mail.warn -/var/log/mail.warn
mail.err /var/log/mail.err
```



---

## 6. LOGGING - rsyslogd -

---

# Logging for news system

news.crit /var/log/news/news.crit

news.err /var/log/news/news.err

news.notice -/var/log/news/news.notice

# Some 'catch-all' logfiles

\*.=debug;auth,authpriv.none;news.none;mail.none -/var/log/debug

\*.=info;\*.=notice;\*.=warn;auth,authpriv.none;cron,daemon.none;\  
mail,news.none -/var/log/messages

# Emergencies are sent to everybody logged in.

\*.emerg \*

---

## 6. LOGGING - rsyslogd -

---

# Per enviar-ho a una tty:

```
daemon,mail.*;news.=crit;news.=err;news.=notice;\  
*.=debug;*.=info;*.=notice;*.=warn /dev/tty6
```

# Per enviar a la cònsola, primer executar:

```
# $ xconsole -file /dev/console [...]
```

# Després

```
daemon.*;mail.*;news.crit;news.err;news.notice;*.=debug;*.=info;\  
*.=notice;*.=warn |/dev/xconsole
```

---

## 6. LOGGING - autres exemples -

---

# The following lines would log all messages of the facility `mail` except those with the priority `info` to the `/usr/adm/mail` file. And all messages from `news.info` (including) to `news.crit` (excluding) would be logged to the `/usr/adm/news` file.

```
mail.*;mail.!=info /usr/adm/mail
```

```
news.info;news.!=crit /usr/adm/news
```

The following are equivalents:

```
mail.none or mail.!=* or mail.!=debug
```

---

## 6. LOGGING - syslog system calls -

---

```
#include <syslog.h>
```

```
void openlog(const char *ident, int option, int facility)
```

Opens a connection to the system logger for a program. The string pointed to by **ident** is prepended to every message.

```
void syslog(int priority, const char *format, ...);
```

Generates a log message.

```
void closelog(void);
```

Close the connection to the system logger for a program.

---

## 6. LOGGING - syslog system calls -

---

**option** The option argument to `openlog()` is an OR of any of these:

`LOG_CONS` Write directly to system console if there is an error while sending to system logger.

`LOG_ERROR` Print to `stderr` as well

`LOG_PID` Include PID with each message

----- autres menys importants

---

## 6. LOGGING - syslog system calls -

---

**facility** used to specify what type of program is logging the message:

LOG\_AUTH (DEPRECATED Use LOG\_AUTHPRIV instead)

LOG\_AUTHPRIV security/authorization messages (private)

LOG\_CRON clock daemon (cron and at)

LOG\_DAEMON system daemons without separate facility value

LOG\_FTP ftp daemon

LOG\_KERN kernel messages

LOG\_LOCAL0 through LOG\_LOCAL7 reserved for local use

LOG\_LPR line printer subsystem

LOG\_MAIL mail subsystem

LOG\_NEWS USENET news subsystem

LOG\_USER (default) generic user-level messages

---

## 6. LOGGING - syslog system calls -

---

**priority** determines the importance of the message:

LOG\_EMERG **system is unusable**

LOG\_ALERT **action must be taken immediately**

LOG\_CRIT **critical conditions**

LOG\_ERR **error conditions**

LOG\_WARNING **warning conditions**

LOG\_NOTICE **normal, but significant condition**

LOG\_INFO **informational message**

LOG\_DEBUG **debug-level message**

---

## 7. Usuaris i Grups

---

- Fitxers de configuració d'usuaris:

`/etc/passwd`

`/etc/shadow`

- Fitxer de configuració de grups:

`/etc/group`



---

## 7. Usuaris i Grups - Fitxer /etc/passwd

---

- Camps (separats per “:”):

Login name

Optional encrypted password

Numerical user ID

Numerical group ID

User name or comment field

User home directory

User command interpreter

- Exemple:

```
francesc:x:1000:1000:Francesc Solsona:/home/francesc:/bin/bash
```

---

## 7. Usuaris i Grups - Fixer /etc/shadow

---

- Camps (separats per “:”):

Login name

Encrypted password

Days since Jan 1, 1970 that password was last changed

Days before password may be changed

Days after which password must be changed

Days before password is to expire that user is warned

Days after password expires that account is disabled

Days since Jan 1, 1970 that account is disabled

Reserved field

- Exemple:

```
francesc:$1$Me/cGKsG$5ui/abvo44aqeY9BF790c0:12430:0:99999:7:::
```

---

## 7. Usuaris i Grups - Fitxer `/etc/group`

---

- Camps (separats per “:”):

name of the group.

the (encrypted) group password.

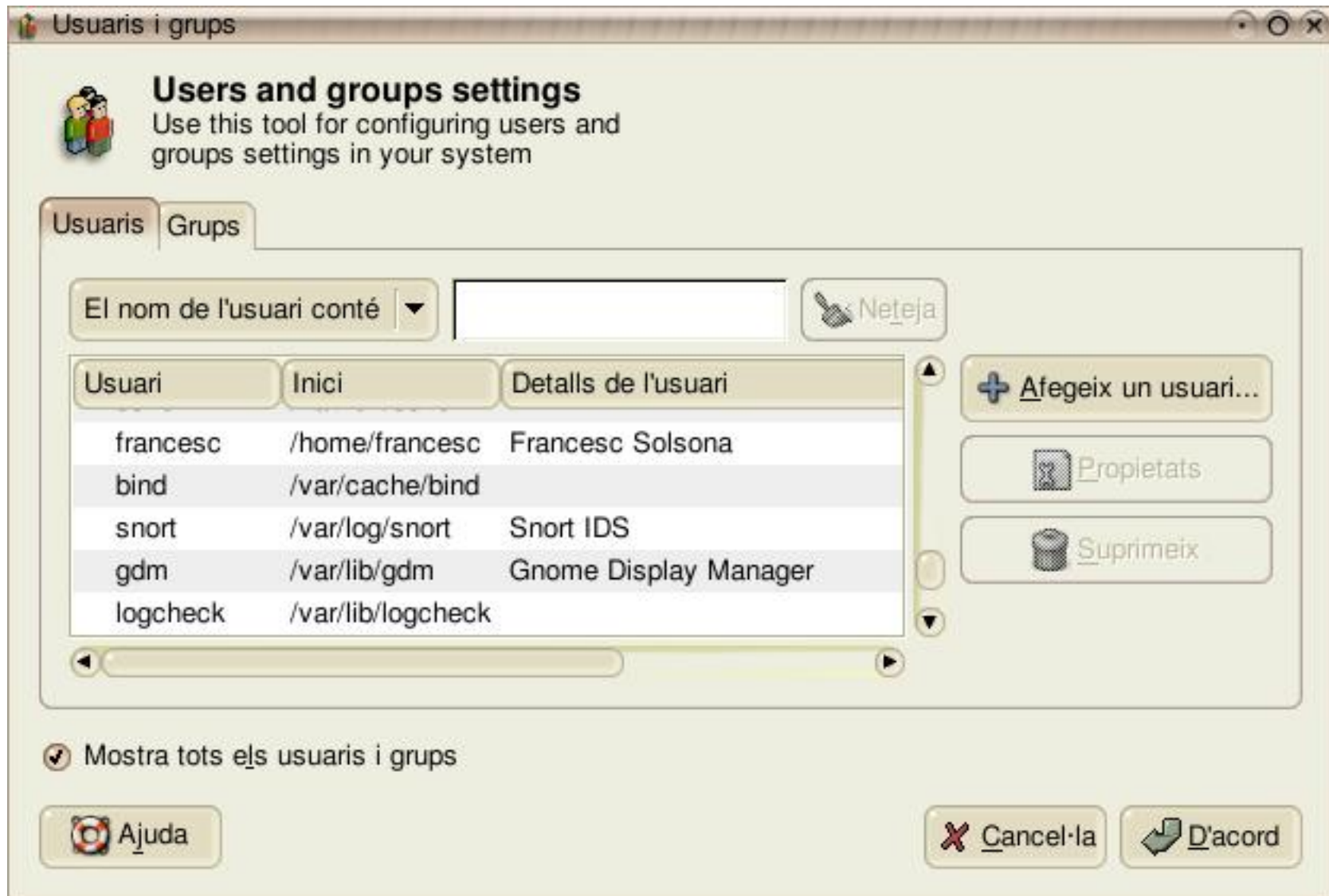
numerical group ID.

group member's user names, separated by commas.

- Exemple:

```
.....  
sudo:x:27:francesc, joan, sara  
audio:x:29:francesc, pep  
video:x:44:francesc  
.....
```

## 7. Usuaris i Grups - Aplicació users-admin



---

## 7. Usuaris i Grups - Possibles aplicacions users-admin (sudo)

---

- Si executem com un usuari normal:

```
$ poweroff
```

```
poweroff: must be superuser
```

- El sistema ens indica que no tenim permisos suficients per executar la comanda `poweroff`
- La comanda `sudo` ens permet executar als usuaris normals aplicacions amb permisos de superusuari. Per exemple:

```
$ sudo poweroff
```