



# LPIC-1. Examen 101. Objectiu 103.1

LPIC-1. Examen 101  
103.1. Treballar a la línia d'ordres

## LPI 103.1. Treballar a la línia d'ordres Wiki:

[http://acacha.org/mediawiki/index.php/LPI\\_103.1](http://acacha.org/mediawiki/index.php/LPI_103.1)

# Objectius

## 103.1. Treballar a la línia d'ordres



- **Objectiu:** Els candidats han de ser capaços d'interactuar amb l'interpret d'ordres (shell) i les comandes que utilitzen la línia d'ordres. Aquest objectiu assumeix que l'interpret d'ordres a utilitzar és bash.
- **Pes:** 4



### Àrees Clau de Coneixement:

- Utilitzar ordres bàsiques de l'interpret d'ordres embolcall i seqüències d'instruccions d'una línia per a fer tasques bàsiques amb la línia d'ordres.
- Utilitzar i modificar l'entorn de l'interpret d'ordres incloent la definició, la referència i l'exportació de les variables d'entorn.
- Utilitzar i editar l'històric d'ordres.
- Invocar ordres dins i fora del camí (path) definit.



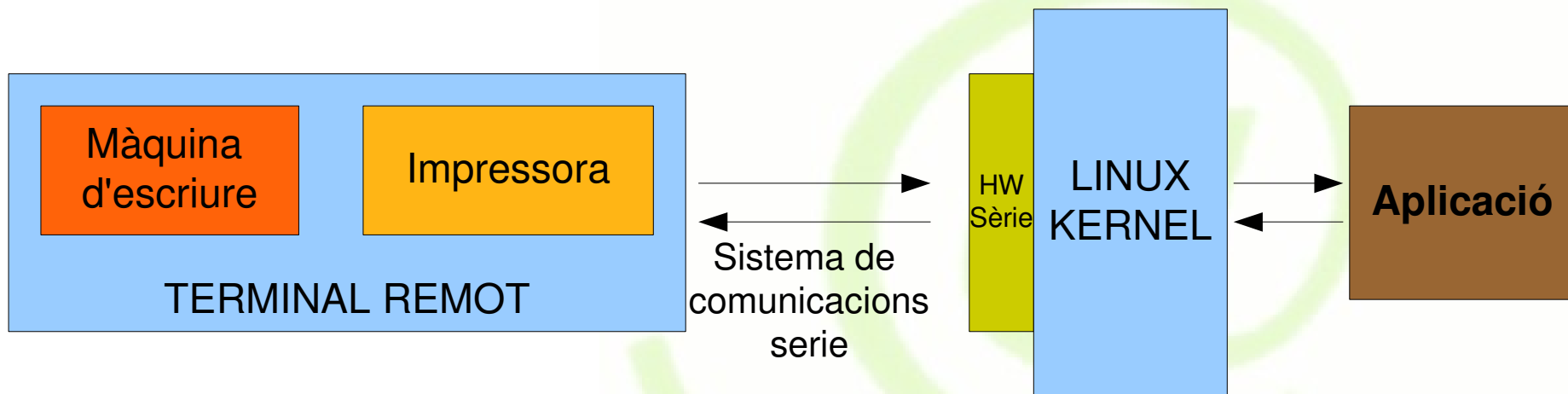
La següent és una llista parcial de fitxers, termes i utilitats utilitzades:

- .
- [bash](#)
- [echo](#)
- [env](#)
- [exec](#)
- [export](#)
- [pwd](#)
- [set](#)
- [unset](#)
- [man](#)
- [uname](#)
- [history](#)



**Apunts:** [LPI 103.1](#). Treballar a la línia d'ordres

## ◆ Concepte de terminal a UNIX/Linux



- ◆ A nivell de maquinari és un sistema obsolet
- ◆ A nivell conceptual és un model encara vàlid adaptat a les complexitats del diferent maquinari actual (teclats, ratolins, llapis, pantalles, pantalles tàctils...)

# Terminals

- ♦ També coneguts com consoles
- ♦ **Tipus de terminals:**
  - ♦ **Terminals de maquinari:** teletipus (tty teletype) i terminals “tontos”
  - ♦ **Consoles virtuals**
  - ♦ **Emuladors de terminal**
  - ♦ **Terminals remots**

# tty. Teletype

## ♦ tty:

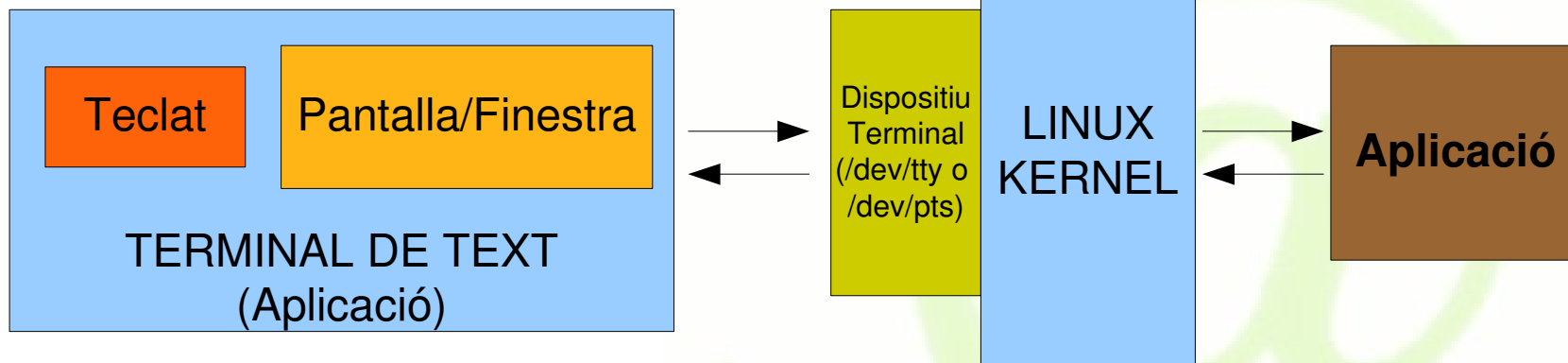
- ♦ Abreviació de **teletype**
- ♦ Els teletypes o teleprinters eren els primer terminals de maquinari.

## ♦ Els dispositius de terminal de Linux encara s'anomenen tty:

```
$ ls -la /dev/tty* | more
crw-rw-rw- 1 root  root    2007-05-13 19:22 /dev/tty
crw-rw---- 1 sergi sergi   2007-05-13 19:26 /dev/tty0
crw----- 1 sergi tty      2007-05-13 19:27 /dev/tty1
crw-rw---- 1 sergi sergi   2007-05-13 19:26 /dev/tty10
.....
crw----- 1 sergi tty      2007-05-13 17:36 /dev/tty2
.....
crw----- 1 bego  tty      2007-05-13 17:37 /dev/tty3
```



## ♦ Model actual



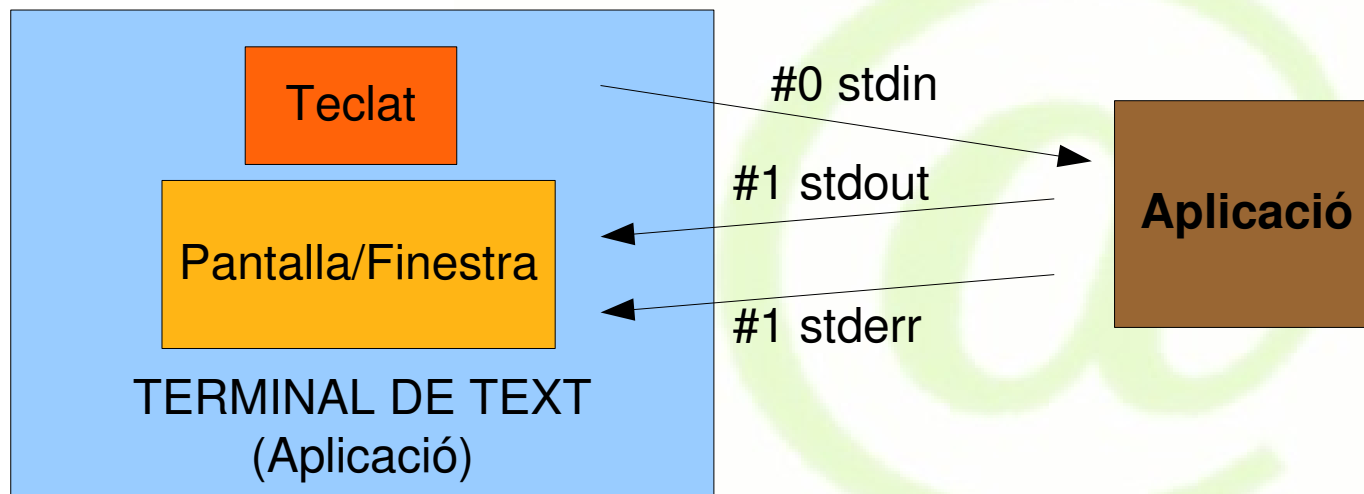
♦ Bàsicament utilitzem terminals basades en la combinació de teclat+ pantalla:

- Terminals virtuals (/dev/tty\*)
- Pseudo Terminals (/dev/pts/\*)



# Descriptors de fitxers terminals

## ◆ Descriptors de fitxers



```
$ lsof | grep pts
bash      20198  sergi    0u      CHR      136,0    2  /dev/pts/0
bash      20198  sergi    1u      CHR      136,0    2  /dev/pts/0
bash      20198  sergi    2u      CHR      136,0    2  /dev/pts/0
bash      20198  sergi   255u    CHR      136,0    2  /dev/pts/0
lsof      21074  sergi     0u      CHR      136,0    2  /dev/pts/0
lsof      21074  sergi     2u      CHR      136,0    2  /dev/pts/0
grep      21075  sergi     1u      CHR      136,0    2  /dev/pts/0
grep      21075  sergi     2u      CHR      136,0    2  /dev/pts/0
```

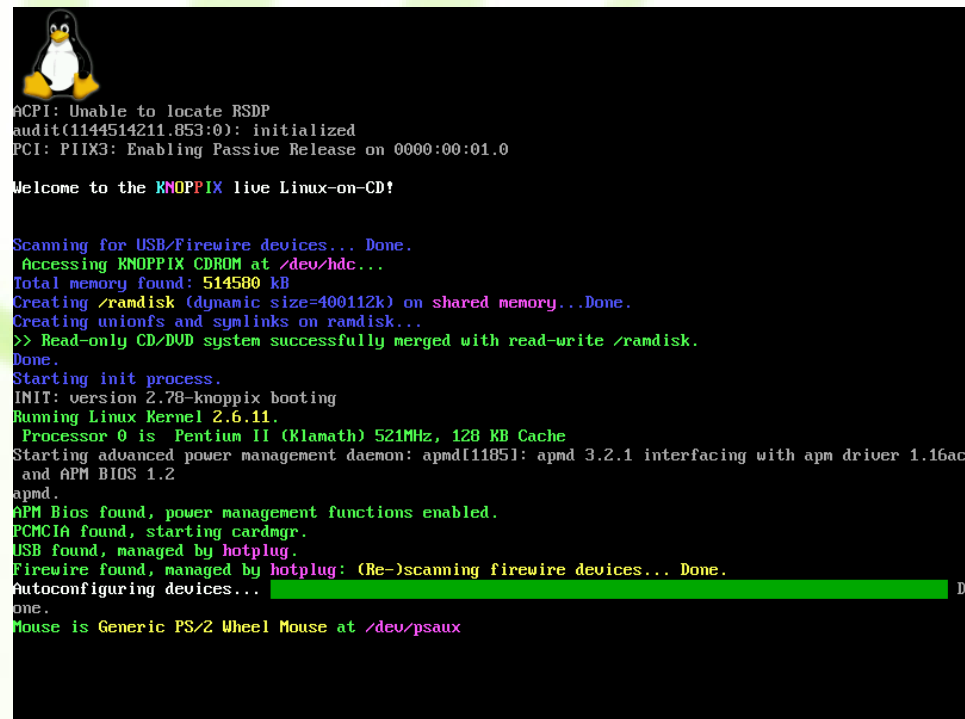


# Terminals Virtuals

## ♦ Virtual console (VC)

- ♦ És la combinació conceptual de teclat+pantalla que creen una interfície d'usuari
- ♦ A Linux normalment tenim 6 consoles virtuals (iniciades pel procés init segons el fitxer (/etc/inittab) o upstart.

Per canviar entre consoles virtuals executem (Ctrl+Alt+F1 o F2 o F3...)



```
ACPI: Unable to locate RSDP
audit(1144514211.853:0): initialized
PCI: PIIX3: Enabling Passive Release on 0000:00:01.0

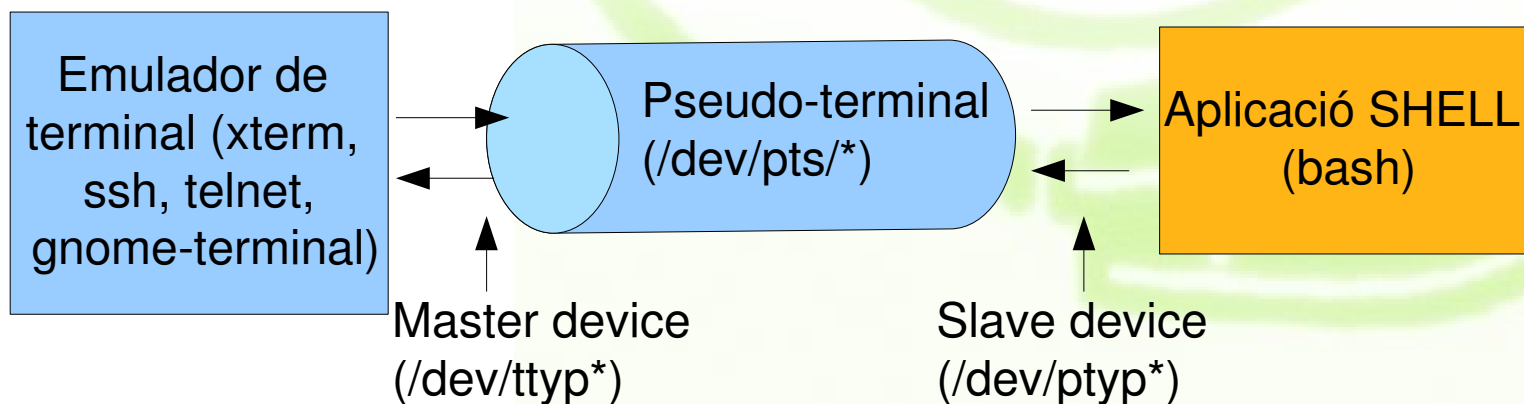
Welcome to the KNOPPIX live Linux-on-CD!

Scanning for USB/Firewire devices... Done.
Accessing KNOPPIX CDRom at /dev/hdc...
Total memory found: 514580 kB
Creating /ramdisk (dynamic size=400112k) on shared memory...Done.
Creating unionfs and symlinks on ramdisk...
>> Read-only CD/DVD system successfully merged with read-write /ramdisk.
Done.
Starting init process.
INIT: version 2.78-knoppix booting
Running Linux Kernel 2.6.11.
Processor 0 is Pentium II (Klamath) 521MHz, 128 KB Cache
Starting advanced power management daemon: apmd[11851]: apmd 3.2.1 interfacing with apm driver 1.16ac
and APM BIOS 1.2
apmd.
APM Bios found, power management functions enabled.
PCMCIA found, starting cardmgr.
USB found, managed by hotplug.
Firewire found, managed by hotplug: (Re-)scanning firewire devices... Done.
Autoconfiguring devices... Done.
Mouse is Generic PS/2 Wheel Mouse at /dev/psaux
```



## ♦ Unix Pseudo-Terminals

- ♦ Són pseudo-dispositius que proveïxen d'una interfície de text sense un consola virtual associada.
- ♦ Un procés s'encarrega de substituir el maquinari.
- ♦ És una forma especial de comunicació entre processos (IPC)



# Emuladors de terminal

## ♦ Emuladors de terminal

- ♦ Aplicació que emula un video-terminal dins d'una altre arquitectura de pantalla (p. ex. X-Windows)
- ♦ Normalment és sinònim de línia de comandes tot i que el terme terminal inclou tots els tipus de terminals remots incloent les interfícies gràfiques

## ♦ Gnome-Terminal



# Emuladors terminals

- ♦ **xterm**: xterm és el terminal per defecte del sistema X-Window. Aquest terminal està instal·lat per defecte en qualsevol Entorn d'escriptori.
- ♦ **Konsole**: Konsole és la terminal per defecte a l'entorn d'escriptori KDE. Konsole està integrat amb altres aplicacions de KDE com Konqueror, Krusader, Kate, Dolphin o Kdevelop.
- ♦ **Gnome Terminal**: Tal com indica el seu nom es tracta de la terminal per defecte de l'entorn d'escriptori Gnome.
- ♦ Eterm, rxvt, mrxvt, term, aterm, Terminal (Xfce)...

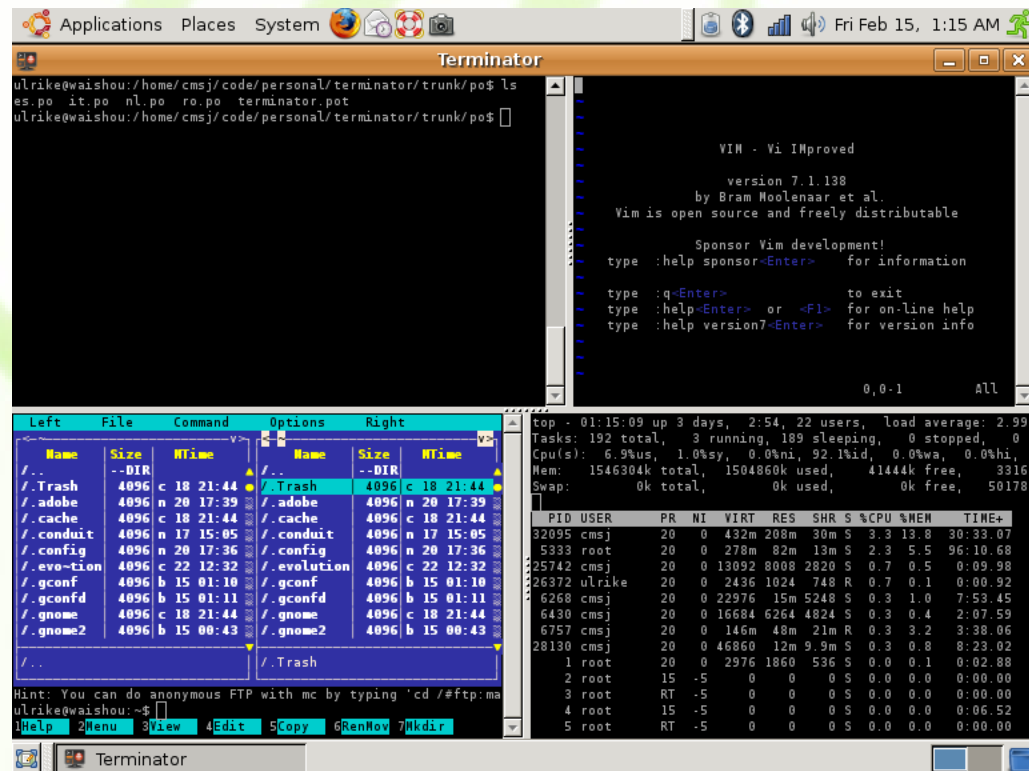
# Emuladors de terminal

## ♦ Terminator



- ♦ Terminal força potent que permet, entre d'altres, dividir la terminal en subfinestres verticals i/o horitzontals

```
$ sudo apt-get install terminator
```



## ♦ Intèrprets d'ordres:

- ♦ **Bash**: bourne Again Shell. El més popular i l'utilitzat en aquest curs.
- ♦ **Sh** (Bourne Shell): un dels primers intèrprets de comandes. Shell per defecte a Unix. Creada al 1977 per Stephen Bourne dels laboratoris AT&T de Bell. sh és encara l'interpret de comandes per defecte de root en sistemes Unix.
- ♦ **csh**: La shell original de C no es gaire utilitzada a Linux. **tcsh** és un bon substitut de csh a Linux, força popular però gairebé cap distribució Linux la porta per defecte
- ♦ **Ksh** (Korn shell): va ser dissenyat per tal de tenir les millors característiques de sh i zsh.
- ♦ **Dash** (Debian Almquist Shell) una versió millorada d'ash. Més petit i eficient que bash. A Ubuntu, des de la versió 6.10, és l'interpret de comandes per defecte del sistema, tot i que l'intèrpret per defecte dels usuaris continua sent bash.

# Però, que és un ordre?

## ♦ El que escrivim després del PROMPT a l'interpret d'ordres

- ♦ El format d'una ordre acostuma a ser:

```
ordre parametre1 parametre2 parametre3 ...
```

- ♦ Els paràmetres poden ser opcions o informació que necessita l'ordre
- ♦ Hi han 3 sistemes (estils) per a les opcions
  - Unix98 : Sistema de paràmetres d'un sol caràcter precedit de guió
  - GNU: proposa que les opcions siguin paraules (--palabra)
  - BSD--> Lletres sense guió

# Però, que és un ordre?

## ♦ Examples

### ♦ Unix98 :

```
$ netstat -puta = $netstat -p -u -t -a = $ netstat -tapu
```

### ♦ GNU: `$ls -all` = `$ls -a` diferent de `$ls -a -l -l`

### ♦ FreeBSD

- L'ordre ps accepta el diferents tipus:

```
$ ps aux
```

```
$ ps -aux
```



# Sintaxi d'un ordre

## ♦ Durant el curs i als manuals de Linux parlarem de la sintaxi de les ordres

- ♦ Per exemple: `cp [OPTION]... SOURCE... DIRECTORY`
- ♦ Els paràmetres que són opcionals es marquen entre claudàtors. Els obligatoris no
- ♦ Els tres punts indiquen que és possible repetir l'opció precedent (en el exemple, es poden repetir opcions i orígens -SOURCE)
- ♦ La sintaxi pot ser més concreta:

```
man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m system[,...]] [-M path] [-S list] [-e extension] [-i|-I] [--regex|--wildcard][--names-only] [-a] [-u] [-P pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z] [[section] page ...] ...
```

# Comandes útils

## ♦ **tty:**

```
$ tty
/dev/pts/1
```

```
$ tty
/dev/tty1
```

- ♦ Ens mostra quina terminal estem utilitzant

## ♦ **getty**

```
$ cat /etc/inittab
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
```

- ♦ Utilitzada pel procés **init** per crear les terminals virtuals i executar login.

## ♦ **finger**

```
$ finger
```

Login	Name	Tty	Idle	Login Time	Office
sergi	Sergi Tur Badenas	*:0		May 17 07:58	
sergi	Sergi Tur Badenas	pts/1	37	May 17 07:59	(:0.0)
sergi	Sergi Tur Badenas	pts/0		May 17 11:14	(:0.0)

- ♦ Mostra informació dels usuaris i les terminals

## ♦ **Recursos: Comandes de terminals a la wiki**

# Comandes útils

## ♦ who

```
$ who -a
      system boot  2007-05-17 07:54
      nivll exec 2 2007-05-17 07:54      darrer=
LOGIN  tty4        2007-05-17 07:54      4422 id=4
LOGIN  tty5        2007-05-17 07:54      4423 id=5
LOGIN  tty2        2007-05-17 07:54      4426 id=2
LOGIN  tty3        2007-05-17 07:54      4428 id=3
LOGIN  tty1        2007-05-17 07:54      4429 id=1
LOGIN  tty6        2007-05-17 07:54      4430 id=6
sergi  ? :0        2007-05-17 07:58      ?      6155
sergi  + pts/1     2007-05-17 07:59 00:28      6516 (:0.0)
sergi  + pts/0     2007-05-17 11:14      .      6516 (:0.0)
```

- ♦ Similar a finger. Mostra també les terminals virtuals
- ♦ **Comandes exit, clear (Ctrl+I) i reset**
- ♦ **Comunicacions entre terminals**
  - ♦ write, msg i talk

# Variable \$TERM

## ♦ Variable d'entorn \$TERM

- ♦ Ens mostra el tipus de terminal que estem utilitzant.

- ♦ X-Windows:

```
$ echo $TERM  
xterm
```

- ♦ Terminal virtual (Ctrl+Alt+F1)

```
$ echo $TERM  
linux
```

# Shell per defecte del sistema

/bin/sh

- ◆ És un enllaç simbòlic a la shell per defecte del sistema (per-system).

```
$ ls -la /bin/sh  
lrwxrwxrwx 1 root root 4 2009-04-26 10:41 /bin/sh -> dash
```

- ◆ Més endavant veurem que cada usuari pot tenir la seva shell per defecte (login shell)
- ◆ Shell per defecte d'un script executable (shebang)

```
$ cat /etc/init.d/apache2 | more  
#!/bin/sh -e  
### BEGIN INIT INFO  
...  
ENV="env -i LANG=C PATH=/usr/local/bin:/usr/bin:/bin"  
...
```

- ◆ Podem canviar d'interpret executant una ordre amb el mateix nom que l'interpret (\$ sh o \$ bash)

# PROMPT

## ♦ to prompt...

- ♦ en anglès vol dir preguntar, sol·licitar, indicar...
- ♦ En informàtica el PROMPT és el text que ens mostra l'interpret d'ordres cada vegada que volem executar una nova ordre. Per exemple: `$`
- ♦ Caràcters habituals de final del PROMPT
  - **Dollar (\$):** usuari corrent
  - **Coixinet (#):** superusuari (root).
- ♦ Una de les combinacions de PROMPT més utilitzades:

```
usuari@nom_maquina:PATH$
```

# Trucs i ús de la línia d'ordres

## ◆ Quines eines tenim que ens permeten fer més àgil i menys tediosa la terminal?

### ◆ Command-completion

- Tabulador: ens permet auto-completar les ordres. Exemple:

```
[nene@lin2 x]$
```

- **Bip:** So que fa l'autocompletar quan hi ha més d'una opció --> tornar a tabular
- Hi ha suport per a:
  - Completar comandes
  - Completar noms de fitxers
  - Algunes ordres també tenen suport per als paràmetres (p. ex. apt-get)



## ♦ Paquet **bash-completion** (Sistemes Debian)

- ♦ Aporta la capacitat d'autocompletar extra per a certes comandes del sistema.
- ♦ Fitxer **/etc/bash\_completion**. Completa algunes comandes com p. ex. apt-get

```
$ cat /etc/bash_completion | grep apt
# Debian apt-get(8) completion.
have apt-get &&
__apt_get()
...
```

- ♦ Altres aplicacions: carpeta **/etc/bash\_completion.d**

```
$ ls /etc/bash_completion.d
acroread.sh  apache2.2-common  apt-file  debconf
desktop-file-validate  dkms  ooffice.sh  pon
rdiff-backup  subversion
```

# Edició amb bash

## ♦ S'utilitza la sintaxi del editor de text emacs

### ♦ Moure's per la línia

- Ctrl+A: inici de línia (també tecla “Inicio”) i Ctrl+E: final de línia (també tecla “Fin”).
- Fletxes esquerra i dreta del teclat. Ctrl+Fletxes ens permet moure'ns a dreta i esquerra de paraula en paraula.

### ♦ Esborrar text

- Ctrl+D (tecla suprimir) i Backspace
- Ctrl+k: Esborra des del cursor fins al final de línia.
- Ctrl+x + Backspace: Esborra des del cursor fins al final de línia.

### ♦ Intercanviar text

- Ctrl+t: intercanvia el caràcter d'abans del cursor pel de després del cursor.
- Canviar de majúscules a minúscules:
  - Esc+U: Converteix del cursor al final de paraula en majúscules
  - Esc+L: Converteix del cursor al final de paraula en minúscules
  - Esc+C: Converteix la primera lletra (Capitalize) de la paraula sota el cursor a majúscules

# Edició amb bash

## ◆ Invocar un editor

- Es pot executar un editor complet per tal d'editar una línia de l'interpret d'ordres.
- Ctrl+X+Ctrl+E. S'executa l'editor definit a la variable d'entorn EDITOR o FCEDIT

## ◆ Hi ha múltiples dreceres de teclat

- ◆ Ctrl+c: Envia la senyal d'acabar el que s'esta executant
- ◆ Ctrl+z: Atura (pausa) un ordre
- ◆ Ctrl+l: netejar la pantalla (equivalent a l'ordre clear)
- ◆ Ctrl+r: Cerca ordres. Consulteu la següent transparència
- ◆ Consulteu la [wiki del curs](#)
  - %27ordres#Dreceres\_.28Shortcuts.29

# Historial de l'interpret de comandes

- ♦ **Bash (i altres shells) mantenen un historial de les ordres executades:**
  - ♦ **Tecla fletxa amunt:** repetir la última ordre executada
  - ♦ Prement múltiples cops la tecla podem anar a ordres més antigues. La tecla fletxa abaix permet anar a la següent ordre en l'historial
  - ♦ Cercar comandes a l'historial (**ctrl+r**):

```
$  
(reverse-i-search) `':
```

## ♦ Permet treballar amb l'històric d'ordres

- ♦ Com funciona? Utilitza dos memòries
  - Memòria temporal (per a cada sessió)
  - Fitxer ~/.bash\_history
- ♦ Cada cop que s'inici una sessió s'omple la memòria temporal amb el contingut de ~/.bash\_history i quan es finalitza és fa a l'inversa.
- ♦ El fitxer ~/.bash\_history no està sempre sincronitzat
- ♦ Cada pantalla d'emulador de terminal té la seva pròpia memòria temporal.
- ♦ Es tracta d'una ordre que forma part de bash (builtin-command)
- ♦ History a la wiki del curs

# Ordre history

## ♦ Consultar l'històric d'ordres

```
$ history
 1  cd carpeta/
 2  ls
 3  cd UbuntuServer
...
```

- ♦ Normalment l'històric guarda les últimes 500 ordres

## ♦ Mostrar les últimes 5 entrades

```
$ history 5
507  vi drblpush.diff
508  ls
509  history 1
510  history 2
511  history 5
```

# Ordre history

## ◆ **Expansors:**

```
!!--> Última ordre  
!n--> Executa l'ordre n de la història d'ordres  
!-n--> Executa l'ordre -n  
!paraula--> Executa la última ordre que contenia  
la paraula indicada.  
Ctrl+P ---> Executa la última ordre.
```

## ◆ **No deixar rastre ;-):**

```
$ history -c (a cada sessió de terminal oberta)  
$ rm ~/.bash_history
```

## ◆ **Esborrar només una entrada:**

```
$ history -d 34
```

## ◆ **Sincronitzar**

◆ Sincronitzar cap al fitxer:

```
$ history -a
```

◆ Sincronitzar des del fitxer:

```
$ history -n
```

◆ Sincronitzar sempre:

```
export PROMPT_COMMAND="history -n; history -a"
```



# Ordre history

- ♦ **Afegir la data d'execució de les ordres:**

```
$ export HISTTIMEFORMAT="%F %T "
```

- ♦ **Canviar la mida del històric en memòria:**

```
$ export HISTSIZE=10000
```

- ♦ Canviar la mida del fitxer ~/.bash\_history

```
$ export HISTFILESIZE=10000
```

- ♦ **Executar una ordre i que no es guardi a l'històric**

- ♦ Només cal posar un espai davant l'ordre

# Ordres internes i ordres externes

## ♦ Hi ha dos tipus d'ordres

- ♦ **externes**: són aplicacions independents de l'interpret d'ordres i s'executen indicant el seu camí complet o directament si es troben a la variable d'entorn PATH.
- ♦ **internes**: són ordres que són proporcionades pel propi intèrpret de comandes. També se les anomena **built-in commands**. Per aquestes ordres no trobarem una executable específic al nostre sistema de fitxers, ja que és el propi executable de l'interpret de comandes qui les proporciona. (cd, pwd, echo...)
  - Podeu consultar una llista de les ordres internes a la [wiki del curs](#) (o a \$ man bash)

## ♦ PATHS

- ♦ NO en tenen (de propi)!

```
$ whereis cd  
cd: /usr/share/man/man1/cd.1posix.gz  
$ wich cd
```

- ♦ Algunes ordres tampoc tindran manual (per exemple l'ordre history). Cal consultar el manual de bash:

```
$ man bash
```

- ♦ Algunes internes ordres com cd funcionen amb sudo:

```
$ sudo bash -c 'cd /root/restricted'
```

## ◆ Configuració de bash

- ◆ Com la majoria de programes Linux la configuració es fa mitjançant un fitxer de text pla
- ◆ 2 nivells de configuració
  - **Configuració d'usuari (per-user):** Fitxers ~/.bashrc i ~/.profile (la titlla (~) indica la HOME del usuari)
  - **Configuració de sistema:** Fitxer /etc/profile
- ◆ Aquests fitxers no són més que guions de shell (scripts).

# Variables d'entorn

## ♦ Environment variables

- ♦ Contenen informació sobre l'entorn d'execució:
  - Designen l'aspecte del teu prompt, el teu directori d'usuari, el teu directori de treball, el nom del teu interpret d'ordres, fitxers que has obert, funcions o comandes que has utilitzat...
- ♦ Són utilitzades per la bash i per altres programes

## ♦ Examples

- ♦ Variables predefinides: PATH, EDITOR, PWD, etc..
- ♦ Variables d'entorn definides pels usuaris.

# Variables d'entorn predefinides

## ♦ Algunes de les més conegudes són

Nom	Funció
USER	Nom de l'usuari en el sistema
UID	Identificador numèric de l'usuari
HOME	El directori home de l'usuari
PWD	El directori de treball actual
SHELL	El nom del shell
\$	Identificador del procés que s'està executant
PPID	L'identificador del procés que ha iniciat el procés que s'està executant
?	El codi de retorn de la darrera instrucció executada

# Variables d'entorn

## ♦ Consultar una variable d'entorn

- ♦ Ordre echo
- ♦ Totes les variables comencen per \$ (incloses les d'entorn)

```
$ echo $PATH  
$ env | grep PATH
```

## ♦ Consultar totes les variables d'entorn

- ♦ Ordre env
- ♦ Proporcionada per **coreutils**
- ♦ Es pot utilitzar autocompletar:
  - Escriviu "\$" i tabuleu dos cops.

```
$ env  
...  
TERM=xterm  
SHELL=/bin/bash  
...  
USER=sergi  
...
```



# Variables d'entorn

## ♦ Establir o modificar una variable d'entorn

```
$ SALUTACIO=Hola  
$ export SALUTACIO  
O tot al mateix temps:  
$ export SALUTACIO=hola
```

## ♦ Esborrar una variable d'entorn

```
$ unset SALUTACIO
```

# Variable d'entorn PATH

## ♦ Cada cop que executem un ordre, l'interpret de comandes:

- 1) Primer es comprova si l'ordre és una ordre interna de l'interpret. Sí és així s'executa l'ordre interna.
- 2) Si l'ordre no és un ordre interna, aleshores es busca l'ordre a les carpetes especificades a la variable d'entorn PATH.

```
$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

- Les carpetes amb executables són normalment les especificades per l'estàndard **FHS**

# Variable d'entorn PATH

- ♦ **Cada sistema o usuari pot tenir un PATH diferent**
  - ♦ P. ex. Hi ha diferències entre distribucions:
    - **Debian:** no posa la carpeta /sbin al PATH dels usuaris normals (només al superusuari)
    - **Ubuntu:** Posa /sbin al PATH del usuaris. Que un ordre estigui al path no vol dir que tinguem permisos per executar-la
- ♦ **Modificar el path--> Modificar una variable d'entorn**
  - ♦ Compte en no esborrar les carpetes ja definides al PATH!
  - ♦ Per exemple, per afegir al PATH la carpeta /home/sergi/bin

```
$ export PATH=/home/sergi/bin:$PATH
```

# Directori de treball

## ♦ Definit a la variable d'entorn PWD

- ♦ PWD --> **P**rint **W**orking **D**irectory

```
$ cd  
$ echo $PWD  
/home/sergi
```

- ♦ Tingueu en compte que el PROMPT normalment també us mostra el directori de treball.
  - ♦ Comanda pwd
    - També mostra el directori de treball
- ```
$ pwd  
/home/sergi
```
- Pot ser interna o externa!
    - No fan exactament el mateix. Mostren diferent els enllaços simbòlics.

# Modificar el directori de treball

## ♦ **Ordre cd (Change Directory)**

- ♦ Ens permet modificar el directori de treball, o el que és el mateix, canviar de directori

```
$ cd /home/sergi/Documents
```

- ♦ És equivalent a modificar la variable d'entorn PWD

```
$ export PWD=/home/sergi/Documents
```

- ♦ Exemples:

- Anar a la HOME de l'usuari:
- Anar a l'arrel del sistema:
- Anar al directori pare:
- Anar a un directori (path relatiu)
- Anar al directori anterior i mostrar-lo

```
$ cd | cd ~ | cd $HOME
```

```
$ cd /
```

```
$ cd ..
```

```
$ cd directori1/prova
```

```
$ cd - | cd "$OLDPWD" && pwd
```

# Directoris especials

- ♦ **Hi ha una sèrie de directoris bàsics força utilitzats:**

| Símbol                                          | Significat                                  | Descripció                                                                                                                                                                                                   |
|-------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .                                               | <a href="#">directori de treball.</a>       | El punt sempre especifica el <a href="#">directori de treball</a> . Pot semblar superflu o inútil però té força utilitats. Consulteu per exemple <a href="#">puntbarra</a> .                                 |
| ..                                              | El directori pare                           | Indica un directori més amunt en l'estructura de directoris.                                                                                                                                                 |
| /                                               | El directori arrel (root)                   | És el directori principal i el de més amunt de l'estructura de directoris (el directori arrel no té directori pare). Recordeu que a <a href="#">Unix/Linux</a> no hi ha unitats de disc (com C: o similars). |
| ~ o la <a href="#">variable d'entorn HOME</a> . | Determina la carpeta principal de l'usuari. | Cada usuari (excepte que s'especifiqui el contrari amb l'opció <a href="#">nologin</a> ) té una carpeta pròpia. Aquesta carpeta és anomenada la carpeta HOME (cada) o carpeta d'usuari.                      |

- ♦ Més endavant veurem quina és l'estructura de directoris habitual a Linux (estàndard FHS)

## Punt barra (./)

- ♦ **Heu observat que el directori de treball actual (.) no és al PATH?**
  - ♦ Es pot afegir però no és recomanable per qüestions de seguretat.
  - ♦ Ens podem pensar que estem executant un ordre típica (com ls) i en realitat estar executant un fitxer anomenat ls del nostre directori de treball.
  - ♦ Per executar una aplicació del directori de treball:

```
$ ./programa_o_executable
```
  - ♦ En nom de la web slashdot (slash=punt, dot=punt) està basada en això.

## ◆ Mostra qualsevol text o variable

```
$ echo Hola Mon!  
Hola Mon!
```

- ◆ Serveix per consultar variables (com p. ex. Les d'entorn):

```
$ echo $PATH  
/home/sergi/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
```

- ◆ Els caràcters especials s'han d'escapar amb la contrabarra i utilitzar echo -e. Per exemple per provocar un bip:

```
$ echo -e "\a"
```

- ◆ Wiki del curs



# Exec. Executar ordres externes

## ♦ És similar a executar les ordres directament

```
$ exec oowriter
```

- ♦ L'única diferència és que substitueix el procés en execució pel nou procés.
- ♦ Si ho executeu des de la línia d'ordres, la línia d'ordres es tancarà.

# Executar ordres en segon terme

## ♦ Al executar un ordre com:

```
$ sleep 5
```

- ♦ La terminal queda bloquejada fins que el procés acabi (sleep s'executa i no fa res durant els segons especificats)
- ♦ En aquest cas l'ordre s'executa en primer terme (foreground)
- ♦ Es poden executar ordres amb segon terme (background) amb &

```
$ sleep 5&  
[1] 3202  
$  
[1]+  Done                  sleep 5
```

# Concatenar ordres. Llistes

## ♦ Tenim diferents operadors per executar múltiples ordres en una mateixa línia

- ♦ Operador AND. Només s'executa la comanda 2 si la primera comanda s'ha executat amb èxit:

```
$ comanda1 && comanda 2
```

- ♦ Operador OR. Només s'executa la segona comanda si la primera torna un codi d'error

```
$ comanda1 || comanda 2
```

- ♦ Llista d'ordres

```
$ ordre1; ordre 2; ordre3; ordre 4
```

## ♦ Expansió per claus

```
a{d,c,b}e --> equival a ade, ace i abe
```

```
/usr/local/src/bash/{old,new,dist,bugs} equival a:
```

```
/usr/local/src/bash/old  
/usr/local/src/bash/new  
/usr/local/src/bash/dist  
/usr/local/src/bash/bugs
```

## ♦ Crear 4 carpetes diferents de cop:

```
$ mkdir /usr/local/src/bash/{old,new,dist,bugs}
```

## ◆ Variables

```
{NOM_VARIABLE}  
tot i que també podem utilitzar:  
$NOM_VARIABLE
```

## ◆ Backsticks (cometes endarrera --> accent obert)

- ◆ Permet substituir el resultat d'una ordre:

```
$ rm `find ./ -name "*~"`  
$ echo `date`  
$ echo $(date)
```

## ◆ Expressions aritmètiques

```
$ echo $((365*24))  
8760
```

- ◆ Més informació al manual de bash (secció EXPANSION)

# Ajuda del sistema

- ♦ **El sistema ens proporciona dues ordres per tal d'obtenir ajuda**
  - ♦ **man (manuals):** ens mostra els manuals de les ordres. Gairebé totes les ordres tenen manual
  - ♦ **info:** algunes ordres prefereixen utilitzar info. Per exemple totes les del paquet coreutils (`$info coreutils` ordre). Si hi ha info també hi ha manual (però a vegades info és més complet)

## ♦ Com funciona man?

```
$ man man
```

- ♦ Per consultar el manual d'un ordre: 

```
$ man nom_ordre
```
- ♦ Recordeu que algunes de les ordres internes no tenen manual propi, sinó que estan explicades dins del manual de l'interpret d'ordres 

```
$ man bash
```
- ♦ man no només mostra informació d'ordres, també ho pot fer de fitxers de configuració, funcions, jocs, crides de sistema...
- ♦ man utilitza l'ordre less per mostrar el manual
  - **Espai**: pàgina següent
  - **Esc+v**: pàgina endarrere
  - **q**: sortir
  - **/text**: cerca el text indicat

```
$ man less
```

## ♦ Seccions:

```
1 --> Programes executables i ordres de l'interpret de comandes
2 --> Crides de sistema proporcionades pel kernel.
3 --> Crides a llibreries proporcionades per programes de llibreries
4 --> Fitxers de dispositiu (normalment emmagatzemats a /dev)
5 --> Formats de fitxers
6 --> Jocs
7 --> Miscel·lània (paquets macro, convencions...)
8 --> Ordres d'administració del sistema (programes que s'executen
    freqüentment o exclusivament per l'usuari root)
9 --> Rutines del kernel
```

♦ En cas de conflicte, les seccions més baixes es mostren primer. Per exemple:

- Ordre passwd:

```
$ man 1 passwd
```

- Fitxer /etc/passwd:

```
$ man 5 passwd
```



## ♦ Seccions d'una pàgina del manual:

- ♦ **NOM:** Nom de la pàgina del manual seguit d'una descripció del contingut.
- ♦ **SINOPSIS:** Sintaxis de la comanda amb les seves opcions i arguments.
- ♦ **DESCRIPCIÓ:** Descripció detallada de la comanda. Hi ha una explicació de les diferents opcions.
- ♦ **EXEMPLES:** Exemples d'ús de la comanda.
- ♦ **VEURE TAMBÉ:** Indica les altres pàgines del manual relatives al tema.
- ♦ **ERRORS:** Conté les observacions respecte a les diferències que es poden trobar entre la comanda GNU de la pàgina consultada i la mateixa comanda en un altre sistema UNIX.

## ◆ Cercar

```
$ man -k network | head
/etc/network/interfaces (5) [interfaces] - network interface configuration for
ifup and ifdown
aireplay-ng (1) - inject packets into a wireless network to generate traffic
aseqnet (1) - ALSA sequencer connectors over network
athstats (8) - simple tool to inspect and monitor network traffic
statistics
avahi-autoipd (8) - IPv4LL network address configuration daemon
byteorder (3) - convert values between host and network byte order
ctstat (8) - unified linux network statistics
dhclient-script (8) - DHCP client network configuration script
display-tele (7) - Forwards the display over a network
dumpcap (1) - Dump network traffic
```

## ◆ L'ordre **apropos** és idèntica a man -k

```
$ apropos
apropos què?
```

# Manuais en castellà

## ♦ A Debian (Ubuntu) es poden instal·lar manuals en castellà:

```
$ sudo apt-get install manpages-es manpages-es-extra
```

- Només es mostren en castellà els manuals que estan traduïts. Els podeu consultar amb: 

```
$ dpkg -L manpages-es
```
- Altres idiomes als repositoris. Paquets manpages-\*
- Comanda whereis ens mostra on està el manual d'una ordre.
- Hi ha manuals extres. Paquets. Manpages-extra, manpages-dev, manpages-posix...

## ♦ Pàgines d'Internet

- <http://manpages.ubuntu.com/>
- <http://linux.die.net/man/>

### Atenció!!

Sí teniu l'entorn en català, haureu de fer:

```
$ export LANG=es_ES  
$ man ping
```

Per poder consultar els manuals en castellà

## ♦ A vegades algunes ordres suporten l'ordre info

- ♦ Típic d'ordres bàsiques i ordres del projecte GNU
- ♦ Compte que a vegades info, en algunes distribucions com Debian o Ubuntu, proporciona la mateixa informació que man tot i que hi ha pàgina info
- ♦ Paquet coreutils

- Incorrecte:

```
$ info chroot
```

- Correcte:

```
$ info coreutils chroot
```

## ◆ Ordres GNU

- ◆ A més de les pàgines del manual i de la documentació info, les comandes GNU accepten habitualment la opció **--help**.
  - Ajuda esquemàtica que descriu la sintaxis general de la comanda i les opcions més utilitzades.

```
$ date --help
Forma d'ús: date [OPCIÓ]... [+FORMAT]
    o bé: date [-u|--utc|--universal] [MMDDhhmm[[CC]AA][.ss]]
Mostra la data actual en el FORMAT indicat, o estableix la data del sistema.

-d, --date=CADENA      Mostra la data descrita a la CADENA en lloc de la data
                        actual («now», ara).
-f, --file=FITXER_DATES
                        Com aplicar «--date» una volta per cada línia de
                        FITXER_DATES.
-r, --reference=FITXER
                        Mostra la data de la darrera modificació del FITXER.
...
```

## ♦ Els manuals i els info no són tutorials ni HOW-TOs

- ♦ Podeu consultar altres fonts d'informació
  - TLPD (<http://tldp.org/>) The Linux Documentation Project.
    - Versió castellana: <http://es.tldp.org/>
  - La documentació de les distribucions. Per exemple, amb Ubuntu:
    - Ubuntu community wiki: <https://wiki.ubuntu.com/>
    - Documentació oficial: <https://help.ubuntu.com/>
    - Ubuntu guide: <http://ubuntuguide.org/>
  - No oblideu la [wiki del curs](#)

## ◆ **Ordre uname**

```
$ uname -a  
Linux sergi-desktop 2.6.28-15-generic #52-Ubuntu SMP Wed Sep 9  
10:49:34 UTC 2009 i686 GNU/Linux
```

- ◆ Hi ha paràmetres per mostrar només part de la informació:
  - `uname -r`: release del kernel
  - `uname -v`: versió del kernel
  - `uname -m`: nom de maquina
  - `uname -s`: sistema operatiu
  - etc...

## ♦ L'ordre interna alias defineix abreviacions d'ordres.

- ♦ Forçar que ls treballi sempre amb colors:

```
$ alias ls='ls --color=auto'
```

- ♦ Podeu consultar els alias que esteu utilitzant executant alias sense paràmetres:

```
$ alias  
alias ls='ls --color=auto'
```

- ♦ Es pot desfer un alias amb unalias:

```
$ unalias rm
```

```
#-----  
# Security.  
#-----  
alias rm='rm -i'  
alias cp='cp -i'  
alias mv='mv -i'  
alias ln='ln -i'
```



# Tancar la sessió

## ♦ **Ordre exit**

- ♦ Tanca la sessió actual. Hi ha dos opcions:
  - **Emuladors de terminals:** tanca la finestra o la pestanya en el cas que es treballi amb múltiples pestanyes
  - **Terminals virtuals:** torna al login. Ho podeu comprovar movent-vos a una terminal virtual amb Ctrl+Alt+F1 o F2 o F3...

## ♦ **Ordre logout**

- ♦ Només s'utilitza a terminals virtuals (equival a exit) o consoles remotes. En emuladors de terminal:

```
$ logout  
bash: logout: not login shell: use `exit`
```



## Reconeixement 3.0 Unported

### Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

### Amb les condicions següents:



**Reconeixement.** Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior  
Això és un resum fàcilment llegible del text legal (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>