



# LPIC-1. Examen 101. Objectiu 103.3

## LPI 103.3. Realitzar tasques bàsiques d'administració de fitxers

**Wiki:**

[http://acacha.org/mediawiki/index.php/LPI\\_103.3](http://acacha.org/mediawiki/index.php/LPI_103.3)

# Objectius

## 103.3. Realitzar tasques bàsiques d'administració de fitxers



- **Objectiu:** Els candidats han de ser capaços d'utilitzar les ordres bàsiques de Linux per a administrar fitxers i directoris.
- **Pes:** 4



### Àrees Clau de Coneixement:

- Copiar, moure i eliminar fitxers i directoris de forma individual.
- Copiar múltiples fitxers i directoris de forma recursiva.
- Esborrar fitxers i directoris de forma recursiva.
- Utilitzar màscares (wildcards) simples i avançades a les ordres.
- Utilitzar find per a localitzar i actuar sobre fitxers en funció del seu tipus, mida o temps.
- Utilitzar tar, cpio i dd.



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





- [cp](#)
- [find](#)
- [mkdir](#)
- [mv](#)
- [ls](#)
- [rm](#)
- [rmdir](#)
- [touch](#)
- [tar](#)
- [cpio](#)
- [dd](#)
- [file](#)
- [gzip](#)
- [gunzip](#)
- [bzip2](#)
- [file globbing](#)



**Apunts:** [LPI 103.3](#). Realitzar tasques bàsiques d'administració de fitxers

# Altres temes inclosos

## 104.6. Crear i canviar els enllaços durs i simbòlics

	<ul style="list-style-type: none"> <li>▪ <b>Objectiu:</b> Els candidats han de ser capaços de crear i gestionar enllaços durs i simbòlics a un fitxer.</li> <li>▪ <b>Pes:</b> 2</li> </ul>
	<p><b>Àrees Clau de Coneixement:</b></p> <ul style="list-style-type: none"> <li>▪ Crear enllaços.</li> <li>▪ Identificar enllaços durs i/o simbòlics.</li> <li>▪ Diferències entre copiar i enllaçar fitxers.</li> <li>▪ Utilitzar enllaços com a ajuda per a tasques d'administració del sistema.</li> </ul>
	<p>La següent és una llista parcial de fitxers, termes i utilitats utilitzades:</p> <ul style="list-style-type: none"> <li>▪ <a href="#">ln</a></li> </ul>
	<p><b>Apunts:</b> <a href="#">LPI 104.6</a>. Crear i canviar els enllaços durs i simbòlics</p>

# Paquet coreutils

- ♦ **La majoria d'ordres de gestió bàsica de fitxers pertanyen a aquest paquet**

- ♦ Està pre-instal·lat a totes les distribucions:

```
$ dpkg -l | grep coreutils  
ii  coreutils  6.10-6ubuntu1
```

The GNU core utilities

- ♦ Abans eren 3 subpaquets
  - **FileUtils**: Les utilitats que veurem en aquest tema
  - **ShellUtils**: Utilitats relacionades amb l'interpret d'ordres
  - **TextUtils**: Les veurem al tema **103.2**
- ♦ **Coreutils a la wiki del curs**

## ♦ A Linux tot són fitxers

- ♦ Un sistema operatiu no és res més que un conjunt de fitxers emmagatzemats en una memòria secundària no volàtil.
  - Normalment es guarda al disc dur (HD). Actualment però també podem trobar sistemes Live-CD/DVD o sistemes emmagatzemats en un llapis de memòria USB.
- ♦ A Linux tot acaba sent un fitxer (fitxers regulars, directoris, unitats de disc, maquinari, sockets, enllaços, conductes...). Carpeta /dev

## ♦ 3 tipologies de fitxers:

- ♦ **Fitxers del Sistema Operatiu:** Fitxers de configuració, executables i aplicacions bàsiques del sistema, etc.
- ♦ **Fitxers d'aplicacions:** Tal i com veurem als temes relacionats amb la gestió de paquets, les aplicacions també no són res més que un conjunt de fitxers que s'instal·len al sistema.
- ♦ **Fitxers d'usuari:** Són aquells fitxers que contenen informació pròpia dels usuaris, com documents de text, fitxers multimèdia, fitxers d'aplicacions, etc.

# Introducció

- ♦ En tots 3 casos, la quantitat de fitxers i tipus de fitxers depèn del tipus de sistema operatiu, de les aplicacions instal·lades i de l'ús que fan del sistema els usuaris.
- ♦ En altres temes (LPI 102.1) es parla sobre com preparar un sistema de fitxers mitjançant el particionat del disc dur mantenint i muntant el sistema de fitxers escollit. Aquest apartat és la continuació lògica pel que fa la gestió de fitxers del sistema
- ♦ Com a administrador, saber gestionar els fitxers del sistema és una competència bàsica. En aquest apartat coneixerem les ordres bàsiques per tal de gestionar fitxers en sistema Linux.



# Noms de fitxer. Conceptes bàsics

## ♦ Noms de fitxers

- ♦ El nom d'un fitxer no és res més que una seqüència o cadena de caràcters (String) que identifiquen de forma unívoca el fitxer dins del sistema.
- ♦ A Linux són força similar a d'altres sistemes operatius

## ♦ Característiques pròpies:

- ♦ Els noms de fitxer en Linux poden contenir qualsevol caràcter excepte:
  - **Barra** (/ forward slash): aquest caràcter està reservat per indicar l'arrel del sistema de fitxers (root) i també és el separador de directoris.
  - **Caràcter null** (ASCII): s'utilitza per indicar el final de segments de text

# Noms de fitxer. Conceptes bàsics

- ◆ Es distingeix entre majúscules i minúscules (case sensitive)
- ◆ Les extensions (tipus .txt), tot i que s'utilitzen per comoditat, no són normatives.
- ◆ Els directoris a Linux són un tipus específic de fitxer i com a tals se'ls hi aplica les mateixes característiques que als "fitxers normals". Cal tenir en compte però que la carpeta arrel sempre és anomenada /.
- ◆ Mida: 1- 255 caràcters (255 bytes).
  - La mida màxima depèn del sistema de fitxers utilitzat, els més habituals (ext2fs, ext3fs, ReiserFS, XFS, etc.) utilitzant com a mida màxima 255. Als primers sistemes operatius Unix la mida es trobava limitada a 14 caràcters.



## ◆ Recomanacions

- ◆ Utilitzar només **caràcters alfanumèrics** (a-z i 1-9) en minúscules
- ◆ **Evitar els espais**: normalment s'utilitza el guió baix ( \_ underscore) com a substitut dels espais o alternativament també el guió simple ( - hyphen) o el punt ( . period)
- ◆ Com a excepció al comentat anteriorment, alguns noms de fitxers com README, INSTALL, NEWS, AUTHORS, LICENSE s'escriuen típicament en majúscules. Aquest fitxers normalment formen part de la documentació de les aplicacions.

# Noms de fitxer. Conceptes bàsics

## ♦ Altres

- ♦ Es permeten les dobles extensions:

```
fitxer_empaquetat_i_comprimit.tar.gz
```

- ♦ Fitxer = conjunt d'informació relacionada entre si
  - Des de la perspectiva del usuari sembla un sol bloc de dades
  - Els noms de fitxer serveixen per tal que els usuaris puguin identificar els fitxers i facilita la tasca de tornar a trobar els fitxers en el futur.
  - Mecanisme similar als **noms de domini d'Internet**. Permet recordar de forma més fàcil l'adreça d'un servidor d'Internet en comptes d'utilitzar adreces IP.
  - Cal recordar, que el sistema operatiu no treballa realment amb els noms de fitxers sinó que utilitza els **inodes**.

## ♦ Sistemes de fitxers FAT i NTFS

- ♦ Es poden muntar a Linux
- ♦ Filesystem type codes
  - msdos: Sistema de noms de fitxers de DOS (8.3)
  - vfat: Noms de fitxers llargs a l'estil de Windows
  - umsdos: extensió només per a Linux que suporta el noms de fitxers a l'estil Linux.
- ♦ Vegeu l'objectiu
  - 104.1. Crear particions i sistemes de fitxers

# Escriure caràcters especials a la shell

## ♦ Com escrivim el nom d'un fitxer amb espais

- ♦ Posant el nom del fitxer entre cometes
- ♦ L'espai és una caràcter especial --> escapar amb \

```
$ mplayer -sub /subtitles/S04E01\ -\ Living\ the\ Dream /torrents/Dexter.S04E01.avi
```

- Altres caràcters com el comodins (-\*,?, etc...) també cal escapar-los
- També podeu utilitzar l'ajuda d'autocompletar noms de fitxers



# ls. Mostrar els fitxers d'un directori

## ♦ ls és una abreviatura de list (l·listar)

- ♦ Ens permet mostrar els fitxers d'un directori. Si executem ls sense paràmetres:

```
$ ls
bin      Documents  examples.desktop  Imatges  Música      Plantilles
Desktop  Escriptori highlight.css      ies4linux-latest.tar.gz  Linkat 3
```

- Ens mostra els fitxers no ocults del directori de treball (recordeu la variable d'entorn PWD)
- La sintaxi de ls és: `$ ls [opcions] [fitxers]`
- Podeu trobar més detalls al manual i a info:

```
$ man ls
$ info coreutils ls
```

## ♦ Paràmetres

- ♦ Mostrar tots (all) els fitxers (ocults): `$ ls -a`
- ♦ Mostrar tots menys . i .. (almost-all): `$ ls -A`
- ♦ Mostrar colors: `$ ls --colors`
- ♦ Mostrar recursivament: `$ ls -r`
- ♦ Mostrar múltiples directoris de cop: `$ ls directori1 directori2 ...`
- ♦ Mostrar més dades dels fitxers (long): `$ ls -l`
- ♦ Mostrar els inodes: `$ ls -i`
- ♦ Mostrar mides en format humà: `$ ls -h`



# ls. Mostrar els fitxers d'un directori

## ◆ Paràmetres

- ◆ Mostrar el sufix (tipus) del fitxer:
- ◆ No mostrar les còpies de seguretat
  - No backup (-B):
- ◆ Mostrar només directoris:
- ◆ Mostrar tota la informació
  - Combinant paràmetres:
- ◆ Mostrar atime timestamp
- ◆ Mostrar ctime timestamp

```
$ ls -F
```

```
$ ls -B
```

```
$ ls -d
```

```
$ ls -Flia
```

```
$ ls -lu
```

```
$ ls -lc
```

# ls. Mostrar els fitxers d'un directori

## Sortida completa

Nombre de blocs del directori

```
sergi@sergi-desktop:~/prova_a_borra$ ls -Flia
total 28
 805122 drwxr-xr-x   5 sergi sergi  4096 2009-10-28 13:28 ./
7585794 drwxr-xr-x 119 sergi sergi 12288 2009-10-28 13:26 ../
 813471 drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 1/
 827943 drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 2/
 829120 drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 3/
 805123 -rw-r--r--   1 sergi sergi    0 2009-10-28 13:26 a
 805124 -rw-r--r--   1 sergi sergi    0 2009-10-28 13:26 b
 805125 -rw-r--r--   1 sergi sergi    0 2009-10-28 13:26 c
 805126 lrwxrwxrwx   1 sergi sergi    1 2009-10-28 13:27 link_to_a -> a
 805127 prw-r--r--   1 sergi sergi    0 2009-10-28 13:28 named_pipe|
```

Inode      Bit      Permisos Fitxer      Usuari Grup      Mida      Marca de temps mtime      Nom fitxer+suffix  
                  Tipus fitxer      Número d'enllaços forts al fitxer

# ls. Mida total

## ♦ La mida total està en blocs

- ♦ Cada bloc 1024bytes-->  $28 \times 1024 = 28672$
- ♦ Compte cada hard links --> No és la mida real.
  - Es poden utilitzar altres eines com du

```
sergi@sergi-desktop:~/prova_a_borra$ ls -la --block-size=1
total 28672
drwxr-xr-x   5 sergi sergi  4096 2009-10-28 17:27 .
drwxr-xr-x 119 sergi sergi 12288 2009-10-28 17:13 ..
drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 1
drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 2
drwxr-xr-x   2 sergi sergi  4096 2009-10-28 13:26 3
-rw-r--r--   2 sergi sergi    0 2009-10-28 13:26 a
-rw-r--r--   1 sergi sergi    0 2009-10-28 13:26 b
-rw-r--r--   1 sergi sergi    0 2009-10-28 13:26 c
-rw-r--r--   2 sergi sergi    0 2009-10-28 13:26 hardlinka
lrwxrwxrwx   1 sergi sergi    1 2009-10-28 13:27 link_to_a -> a
prw-r--r--   1 sergi sergi    0 2009-10-28 13:28 named_pipe
```

# ls. Modificar la sortida

## ♦ Podem mostrar els resultats de ls de múltiples maneres:

```
-S o --sort=size: Ordena per mida de fitxer
-t o --sort=time: Ordena per data de modificació (mtime de
                  L'inode). Els fitxers nous primer
-u o --time=atime: En sortides amb -l ordena per la última data
                  d'accés al fitxer (atime de l'inode)
-U o --sort=none: sense ordre. És útil al mostrar carpetes amb
                  molts fitxers, ja que el fet de no ordenar
                  pot fer més ràpida l'execució de l'ordre.
-v o --sort=version: Ordena per versió
-X o --sort=extension: Ordena per extensió del fitxer.
-r o -reverse:      Ordre invers
```

# ls. Modificar la sortida

- ♦ **Mostrar una sola columna**
- ♦ **Llista separada per comes**

```
$ ls -l
bin
Desktop
Documents
Escriptori
examples.desktop
highlight.css
ies4linux-2.99.0.1
ies4linux-
latest.tar.gz
Imatges
Linkat 3
Música
Musiquetes.pls
Plantilles
prova
```

```
$ ls -m
bin, Desktop, Documents,
Escriptori,
examples.desktop,
highlight.css, ies4linux-
2.99.0.1, ies4linux-
latest.tar.gz, Imatges,
Linkat 3, Música,
Musiquetes.pls, Plantilles,
prova
```

CSV: Comma Separated Values

```
$ ls -mQ
"a", "b", "B", "Ba", "c",
"d", "D"
```

# Tipus de fitxers

## ◆ Colors de ls:

- ◆ La majoria de distros activen els colors per defecte. A ~/.bashrc:

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

Color	Tipus de fitxer
Negre	Fitxer normal
Blau	Directoris
Vermell	Fitxers comprimits
Blanc	Fitxers de text
Rosa	Imatges
Cyan	enllaços
Groc	Fitxers especials de dispositiu
Verd	executables
Vermell que parpadeja	enllaços trencats

## ◆ Sufix (opció -F de ls)

```
Res  Fitxer normal
/    Directori
@    Enllaç simbòlic
=    Endoll ( (Domain) Socket)
|    Conducte (Named pipes)
*    Fitxer executable
```

## ◆ Prefix (opció -l de ls)

```
-    Fitxer normal (Regular File)
d    Directori
l    enllaç simbòlic
c    Fitxer especial. Dispositiu de caràcter
b    Fitxer especial. Dispositiu de bloc
s    endoll ( (Domain) socket)
p    conducte ( Named Pipe)
```



# Tipus de fitxers

## ♦ A Linux tot són fitxers però de diferents tipus:

- ♦ **Fitxers normals (regular files):** són els fitxers normals i corrents (ni directoris, ni enllaços, etc.)
- ♦ **Directoris:** Els directoris a Linux són un tipus especial de fitxers que contenen altres fitxers o directoris.
- ♦ **Enllaços simbòlics:** Són fitxers que apunten a altres fitxers.
- ♦ **Endolls (Sockets):** Alguns fitxers poden ser Unix Domain Sockets. Em podeu trobar exemples a la carpeta /var/run. No els veurem en aquest curs.
- ♦ **Conductes (Pipes):** Són un tipus específic de conductes. Consulteu conductes amb nom.
- ♦ **Fitxers especials de dispositiu:** Normalment són els que trobem a la carpeta /dev i poden ser de dos tipus, dispositius de bloc (b) i dispositius de caràcter (c).

## ♦ Fitxers ocults

- ♦ Els fitxers (o directoris) que comencen per punt (.) són tractats pel sistema com fitxers ocults.
- ♦ No es mostren per defecte en aplicacions com ls o navegadors gràfics com nautilus a no ser que s'indiqui mitjançant paràmetres o opcions de menú concretes.
- ♦ Amb ls, es poden mostrar els fitxers ocults amb -a (all):

```
$ ls -a
drwxr-xr-x  6 root  root          4096 2009-09-10 17:31 ..
drwx----- 5 sergi  sergi          4096 2009-08-23 11:31 .adobe
drwxr-xr-x  3 sergi  sergi          4096 2009-05-15 05:57 .ApacheDirectoryStudio
-rw-----  1 sergi  sergi       198406 2009-10-27 21:33 .bash_history
...
```

# Fitxers especials

- ♦ A diferència d'altres sistemes com DOS (8.3 filename) el punt no s'utilitza només com a separador entre el nom del fitxer i la seva extensió.
- ♦ Podem trobar més d'un punt en el nom d'un fitxer i utilitzar-lo com a substitut de l'espai

**NOTA:** Tingueu en compte que els directoris especials `.` i `..` són un tipus específic de directoris ocults.

- ♦ El directoris ocults són molt utilitzats per tal d'emmagatzemar les configuracions d'usuari a la seva carpeta d'usuari (la HOME).
- ♦ **Fitxers de còpia de seguretat (nom\_fitxer~)**
  - ♦ Algunes aplicacions com nautilus, per defecte no ens mostren aquests fitxers (són tractats com fitxers ocults)
  - ♦ Per conveni, alguns programes realitzen còpies de seguretat automàtiques de fitxers.

# File globbing

## ♦ Caràcters amb significat especial

- ♦ Cal evitar el seu ús en noms de fitxers (o escapar-los amb \)

- ♦ **Asterisc (\*):**

- Caràcter de substitució, comodí o wildcard més utilitzat
- Coincideix amb qualsevol caràcter o conjunt de caràcters:

```
$ ls p*a -> ls patata paa pab pac paac prova persona ...
```

- Els caràcters de substitució són interpretats per l'interpret d'ordres i passat un cop substituïts a l'ordre que estem utilitzant. A aquest procés se l'anomena file globbing.

- ♦ **Interrogant (?):**

- L'interrogant (? question mark) és el comodí que coincideix amb qualsevol caràcter (un del sol).

```
$ ls p?a -> ls paa pba pca pda pea pfa pwa p~a p½a...
```

## ♦ Valors entre claudàtors:

- Amb els claudàtors podem indicar rangs de caràcters
  - Combinacions OR. Per exemple: p[ao]p
- També és possible indicar rangs amb l'ús del guió. Per exemple: p[a-z]p
  - Diferent de p?p. Només caràcters de l'abecedari en minúscules

## ♦ Barra invertida (\):

- Serveix per escapar caràcters especials. La pròpia contrabarra (\\)
- La barra (/) es reserva com a nom de l'arrel del sistema i com a separador de noms de carpetes.
  - És l'únic caràcter que no podreu utilitzar en un nom de fitxer, la resta es poden utilitzar però és recomanable no utilitzar-los per tal d'evitar confusions.

## ◆ **Ordre touch**

- ◆ S'utilitza per crear fitxers buits: `$ touch noufitxer.txt`
- ◆ Encara que en veritat s'utilitza per modificar les marques de temps (timestamps)
- ◆ Touch, toca el fitxer, i n'actualitza la marca de temps
- ◆ A un fitxer existent: `$ touch fitxer_existent.txt`
  - Per defecte, l'ordre touch estableix la data de modificació i de l'últim accés al temps actual



# Marques de temps

- ♦ **Vitals pel correcte funcionament d'algunes aplicacions. 3 marques de temps:**
  - ♦ **Temps de l'últim accés** (last acces time o **atime**): Marca l'últim cop que el fitxer ha estat **llegit**.
  - ♦ **Temps de l'última modificació** (last modification time o **mtime**): Marca l'últim cop que els continguts del fitxer van ser modificats.
  - ♦ **Temps de canvi** (change time o **ctime**): marca l'últim cop que els permisos del fitxer van ser modificats.
  - ♦ Ja hem vist anteriorment que es poden consultar amb **ls**, però també amb **stat**
    - L'ordre **find** ens permet cercar fitxers segons les seves marques de temps.

# Marques de temps

## ◆ Exemple

- ◆ ls ordena per ordre alfabètic

```
$ touch a
$ touch b
$ touch c
$ touch d
$ touch B
$ touch Ba
$ touch D
$ ls
a  b  B  Ba  c  d  D
```

```
$ ls -l --sort=time
total 0
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:51 D
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:51 Ba
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:51 B
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:51 d
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:50 c
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:50 b
-rw-r--r-- 1 sergi sergi 0 2009-10-11 17:50 a
```

- ◆ Torneu a fer touch alguns fitxers i repetiu ls

- ♦ **Permet consultar de cop els 3 tipus de marques de temps d'un fitxer:**

```
$ stat a
  File: «a»
  Size: 1                      Blocks: 8                      IO Block: 4096
fitxer ordinari
Device: 801h/2049d      Inode: 3721307      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid:
( 1000/   sergi)
Access: 2009-10-11  20:16:24.000000000  +0200
Modify: 2009-10-11  20:16:47.000000000  +0200
Change: 2009-10-11  20:16:47.000000000  +0200
```

# Copiar fitxers. Ordre cp

♦ **La sintaxi de cp és:** `$ cp [opcions] origen destinació`

- ♦ Cp és una abreviatura de **copy**
- ♦ Les carpetes s'ha de copiar amb -r (--recursive o -R)

```
$ cp -r bin Documents
```

- ♦ **IMPORTANT:** El fitxer destí es sobreescriu si ja existeix
- ♦ Mode interactiu (-i): Pregunta abans de sobreescriure
- ♦ Per seguretat, alguns sistemes tenen un alias definit amb -i

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

# Copiar fitxers. Ordre cp

## ◆ Opcions

- ◆ Preservar el propietari (**-p --preserve**)
- ◆ Mantenir els enllaços simbòlics al copiar carpetes
  - Per defecte no es mantenen (es copien els fitxers enllaçats)
    - **-d o --no-dereference (-P) o --preserve=links**
  - Amb fitxers si que es copien correctament
- ◆ Copiar tot el camí complet (**-r --parents**)
- ◆ Copiar només si el fitxer origen és més nou que el de destinació (**-u o --update**)
- ◆ Mostrar més informació (**-v o --verbose**)
- ◆ Fer una còpia exacte (**-dpR**)

# Moure i renombrar fitxers (mv)

## ♦ Sintaxi:

```
$ mv [opcions] origen destinacio
```

- ♦ mv és una abreviatura de **move**
- ♦ A Linux no hi ha una ordre per canviar el nom d'un fitxer. S'utilitza mv.
- ♦ Exemple:
  - Observeu la / final. És obligatòria? No però...
  - ... serveix per assegurar-se que la destinació és una carpeta i no pas un fitxer. mv com cp sobreescriu la destinació
  - Si Documents és una carpeta les següent ordres són equivalents:

```
$ mv document.txt /home/sergi/Documents/
```

```
$ mv document.txt /home/sergi/Documents
```



# Moure i renombrar fitxers (mv)

## ♦ L'origen pot ser una carpeta (no cal -r)

- ♦ Si la destinació és una carpeta existent, és mou l'origen a la carpeta de destinació
- ♦ Si la destinació no existeix, es tracta d'un canvi de nom de la carpeta

## ♦ Canviar el nom d'un fitxer

```
$ mv nom_original nom_nou
```

- ♦ Canviar el nom i la carpeta

```
$ mv nom_original nou/cami/nom_nou
```

- Comparteix algunes opcions amb cp (--interactive, --update, --force o --verbose. )
- Altres no s'apliquen (--preserve, --recursive o --archive )

# Com funciona?

## ♦ **Canviant l'inode**

- ♦ No es mouen els blocs de dades al disc dur
- ♦ Només es canvien les referències a aquests blocs
- ♦ Operació molt ràpida
  - Comproveu-ho movent un fitxer i executant l'ordre stat abans i després de moure el fitxer

## ♦ **Excepció**

- ♦ Moure fitxers entre dos sistemes de fitxers diferents (particions). Aleshores primers es copien les dades, i després es borra l'origen. Es pot cancel·lar de forma segura (Ctrl+c)

# Eliminar fitxers

## ◆ **Ordre rm**

- ◆ És una abreviatura de **remove**

- ◆ La sintaxi és: 

```
$ rm [opcions] fitxer1 [fitxer2]
```

- ◆ Com cp, per treballar amb directoris cal l'opció **-r (--recursive)**

```
$ rm -r directori
```

- ◆ Compte que no pregunta si esteu segurs i el fitxer és modificable (no és de només lectura)

```
$ chmod 400 3
$ rm -r 3
rm: voleu eliminar el directori protegit contra escriptura «3»?
```

- ◆ Es pot forçar a que no preguntis amb **-f (--force)**.
  - Al superusuari mai pregunta!

# Operacions amb directoris

## ♦ **Mkdir (make dir)**

- ♦ Permet crear un directori:

```
$ mkdir [opcions] directori [mes directoris]
```

- ♦ Es poden crear múltiples directoris de cop
- ♦ La opció més utilitzada és -p. Crea tots els directoris pare necessaris:

```
$ mkdir -p /carpeta1/carpeta2/carpeta3
```

## ♦ **rmdir**

- ♦ Esborra directoris buits. Per a buits o no buits millor:
  - rm -r

## ♦ Enllaços

- ♦ Permeten que un sol fitxer tingui múltiples camins en un sistema de fitxers.
- ♦ El fitxer **només existeix un cop** i **només ocupa l'espai un cop**, però pot tindre múltiples camins (paths).

Els enllaços són similars (encara que força més potents) als accessos directes de Windows o els alias de Mac OS.

Windows Vista suporta symlinks

## ♦ Utilitats

- ♦ Assignar noms (camins) més simples a fitxers amb noms complicats
- ♦ Assignar diferents noms a un mateix fitxer
- ♦ Algunes ordres que semblen ordres diferents són realment enllaços que porten al mateix executable (per exemple les ordres **fsck**).
- ♦ El sistema operatiu els utilitza de múltiples maneres, per exemple, els guions d'inicialització **System V** o la configuració de sites i mòduls a **Apache** (a2ensite, a2enmod)
- ♦ ...

## ♦ Dos tipus d'enllaços:

- ♦ Enllaços simbòlics o soft links (aka symlinks)
- ♦ Enllaços durs (hard links)

## ♦ **Ordre ln**

- ♦ És l'encarregada de crear enllaços dels dos tipus.
- ♦ Per defecte genera enllaços durs
- ♦ Sintaxi

```
$ ln [opcions] origen enllaç
```



## ♦ Enllaços simbòlics (opció -s o --symbolic)

```
$ mkdir prova  
$ cd prova  
$ touch a  
$ du -h .  
4,0K  .  
$ ln -s a b
```

### ♦ Comproveu el enllaç:

```
sergi@sergi-desktop:~/Documents/images/prova$ ls -l  
total 0  
-rwxr-xr-x 1 sergi sergi 0 2009-10-31 15:26 a  
lrwxrwxrwx 1 sergi sergi 1 2009-10-31 15:26 b -> a
```

- Amb du podeu tornar a comprovar que no augmentat la mida de la carpeta (4K és el mínim que ocupa un fitxer, encara que estigui buit).

## ♦ Propietats

- ♦ Si s'esborra l'enllaç no s'esborra el fitxer original
- ♦ Si s'esborra el fitxer enllaçat aleshores el link no apunta a cap lloc correcte (enllaç trencat)

```
sergi@sergi-desktop:~/Documents/images/prova$ ls -l  
total 0  
lrwxrwxrwx 1 sergi sergi 1 2009-10-31 15:26 b -> a
```

## ♦ Opcions

- ♦ Esborrar els fitxers de destinació (-f o --force): S'elimina l'enllaç si ja existeix. Per defecte, si l'enllaç ja existeix
- ♦ Mode interactiu (-i): Pregunta si s'ha d'eliminar o no els fitxers de destinació existents.

# Enllaços durs

## ♦ Opció per defecte de ln

- ♦ Els enllaços durs són dos noms de fitxer que apunten al mateix inode. Tots dos fitxers (o noms de fitxer) són igual de vàlids, i cap té més importància.
- ♦ L'única diferència és que un s'ha creat primer i l'altre més tard.

Per esborrar un fitxer (inode) amb múltiples enllaços durs cal esborrar tots els enllaços durs.

## ♦ Symlinks vs hardlinks

- ♦ Esborrar un enllaç simbòlic no esborra **mai** el fitxer original.
  - Un enllaç dur només esborra els continguts del fitxer si és l'últim enllaç dur al fitxer.
- ♦ Els enllaços simbòlics són una mica més lents en accés que els enllaços durs.
- ♦ Els enllaços durs no poden apuntar entre fitxers de diferents sistemes de fitxers.
- ♦ Esborrar el fitxer enllaçar per un enllaç simbòlic, deixa l'enllaç simbòlic trencat. Esborrar un dels enllaços durs no afecta a la resta d'enllaços (només decreix el comptador link del inode)
- ♦ No es poden crear enllaços durs de directoris (però si enllaços simbòlics)



# Enllaços durs

## ◆ Exemple

```
$ touch a
$ stat a
File: «a»
Size: 0          Blocks: 0          IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d Inode: 57526     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:04:46.000000000 +0200
$ ln -s a symlink_b
$ stat symlink_b
File: «symlink_b» -> «a»
Size: 1          Blocks: 0          IO Block: 4096   enllaç simbòlic
Device: 801h/2049d Inode: 57528     Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:05:42.000000000 +0200
Modify: 2009-10-12 20:05:41.000000000 +0200
Change: 2009-10-12 20:05:41.000000000 +0200
$ ln a b
$ stat a
File: «a»
Size: 0          Blocks: 0          IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d Inode: 57526     Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:08:29.000000000 +0200
$ stat b
File: «b»
Size: 0          Blocks: 0          IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d Inode: 57526     Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:08:29.000000000 +0200
```

```
$ ls -li
6242474 -rw-r--r-- 2 sergi sergi 5 2008-05-10 11:23 a
6242474 -rw-r--r-- 1 sergi sergi 0 2008-05-10 11:22 b
```

# Enllaços durs

## ♦ Si continuem fent enllaços...

```
$ ln b c
$ stat a
  File: «a»
  Size: 0                Blocks: 0                IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d      Inode: 57526             Links: 3
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:12:27.000000000 +0200
```

```
$ stat b
  File: «b»
  Size: 0                Blocks: 0                IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d      Inode: 57526             Links: 3
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:12:27.000000000 +0200
```

```
$ stat c
  File: «c»
  Size: 0                Blocks: 0                IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d      Inode: 57526             Links: 3
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:12:27.000000000 +0200
```



## ♦ Ara esborrem un harlink

```
$ rm c
```

L'únic que passa es que decrementem el comptador link:

```
$ stat a
  File: «a»
  Size: 0                Blocks: 0          IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d      Inode: 57526       Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:13:31.000000000 +0200
```

```
$ stat b
  File: «b»
  Size: 0                Blocks: 0          IO Block: 4096   fitxer ordinari buit
Device: 801h/2049d      Inode: 57526       Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sergi)   Gid: ( 1000/   sergi)
Access: 2009-10-12 20:04:46.000000000 +0200
Modify: 2009-10-12 20:04:46.000000000 +0200
Change: 2009-10-12 20:13:31.000000000 +0200
```



# Enllaços durs

## ♦ **Aleshores, un harlink no és una copia d'un fitxer?**

- ♦ No és exactament el mateix
  - Els continguts dels fitxer només són emmagatzemats un cop
  - Si es modifica qualsevol dels enllaços durs aleshores es modifiquen tots dos els fitxers.
  - Si es canvia qualsevol de les metadades dels enllaços durs (permisos, propietaris, marques de temps) es modifiquen a tots els fitxers
  - Si s'esborren els continguts d'un dels fitxers s'esborren el de tots, o si es sobreescriu o s'edita un fitxer.

## ♦ Còpia amb link dur

- ♦ **-l**: En comptes de fer una còpia crea un enllaç dur.
- ♦ **-a**: opció archive. Manté els permisos, enllaços, propietaris, etc dels fitxers.
  - **Resultat**: una còpia exacte però sense ocupar dos vegades l'espai original i sense tardar tant en fer la còpia.
  - A la còpia només es poden afegir o eliminar (unlink) fitxers
  - Si modifiqueu la còpia modifiqueu l'original!!!!!!
- ♦ Si ho combinem amb rsync es poden fer **còpies de seguretat incrementals** completes.

# 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). Lo 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)

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

- ◆ Només fitxers regulars pel seu nom (des del directori de treball)

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

- ◆ Només directoris regulars per nom (poder llegir tots els fitxers)

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

# 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 \( -perm -04000 -o -perm -02000 \)`
- ◆ 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
```



# Arxivar i comprimir fitxers

## ♦ Arxivar/Empaquetar

- ♦ L'eina més utilitzada en sistemes lliures és **tar** (Tape **AR**chiver)
- ♦ Eina molt antiga. Inicialment s'utilitzava per fer còpies en cinta
  - Per defecte, tar treballa amb cintes. Per això veureu sempre l'opció -f (indica treball amb fitxers)

Comanda	Comanda curta	Descripció
--create	c	Crea un arxiu tar
--concatenate	A	Afegeix fitxers tar a un fitxer
--append	r	Afegeix fitxers que no són tar a un arxiu ta
--update	u	Afegeix només els fitxers que són nous a un arxiu
--diff o --compare	d	Compara un arxiu amb els fitxers del disc
--list	t	Mostra els continguts d'un arxiu
--extract o --get	x	Extreu els fitxers d'un arxiu

# Arxivar i comprimir fitxers

## ♦ Empaquetar

```
$ tar cf fitxer.tar fitxers
```

- ♦ On fitxers pot ser una carpeta :

```
$ tar cvf fitxer.tar FestaUbuntu
```

- Per defecte ordre silenciosa. -v (--verbose) mostra el que fa l'ordre
- També podem indicar una llista de fitxers (separada per espais) o utilitzar comodins (\*)
- Si el paquet ja existeix, es sobreescriu.

## ♦ Desempaquetar

```
tar -xvf fitxer.tar
```

- ♦ Si els fitxers ja existeixen, es sobreescriuen

# Arxivar i comprimir fitxers

## ♦ **Mostrar els fitxers d'un paquet tar**

```
$ tar tf fitxer.tar  
$ tar tvf fitxer.tar
```

## ♦ **Empaquetar mantenint els permisos**

```
$ tar czfp ile.tar.gz archivewithtar/
```

## ♦ **Comparar. Vegeu l'exemple de la wiki del curs**

```
$ tar czfp ile.tar.gz archivewithtar/
```

## ◆ Compressió de fitxers

- ◆ En sistemes lliures 2 són els sistemes de compressió més utilitzats
  - **Gzip (.gz):** GNU Zip. Algoritme DEFLATE (LZ77 i el Huffman) algorismes lliures sense patents.
    - No s'ha de confondre el gzip amb el Zip ja que no són compatibles.
  - **Bzip2 (.bz2):** Posterior a gzip. Comprimeix la majoria d'arxius de forma més efectiva que els compressors tradicionals gzip o ZIP, però és més lent.
    - Cada cop més utilitzat (Llei de Moore)
- ◆ Cap dels dos sistemes empaqueta (ja tenim tar)

# Arxivar i comprimir fitxers

## ♦ Desempaquetar tarballs

```
$ tar xvzf fitxercomprimit.tar.gz
```

## ♦ Empaquetar tarballs

```
$ tar -czf fitxers.tar.gz fitxers
```

## ♦ Llistar un tarball

```
$ tar -ztvf file.tar.gz
```

## ♦ Comprimir

```
$ gzip fitxer
```

## ♦ Descomprimir

```
$ dpkg -L ifupdown | grep doc | grep interfaces  
/usr/share/doc/ifupdown/examples/network-interfaces.gz  
$ sudo gunzip /usr/share/doc/ifupdown/examples/network-interfaces.gz  
$ vi /usr/share/doc/ifupdown/examples/network-interfaces
```

# Arxivar i comprimir fitxers

## ♦ Desempaquetar tarballs

```
$ tar jxvf fitxercomprimit.tar.bz2
```

## ♦ Empaquetar tarballs

```
$ tar cvjf tarfile.tar.bz2 fitxers..
```

## ♦ Llistar un tarball

```
$ tar -jtvf file.tar.bz2
```

## ♦ Comprimir

```
$ gzip fitxer
```

## ♦ Descomprimir

```
$ dpkg -L ifupdown | grep doc | grep interfaces  
/usr/share/doc/ifupdown/examples/network-interfaces.gz  
$ sudo gunzip /usr/share/doc/ifupdown/examples/network-interfaces.gz  
$ vi /usr/share/doc/ifupdown/examples/network-interfaces
```

## ♦ Com tar, permet arxivar fitxers

### ♦ 3 modes d'operació

- **Mode copy-out (-o o --create-option):** Aquest mode crea un arxiu i copia fitxers a aquest arxiu.
- **Mode copy-in (-i o --extract):** Aquest mode extreu fitxers d'un arxiu. És possible extreure només fitxers concrets o fitxers segons un patró
- **Mode copy-pass (-p o --pass-through):** Combina els dos modes anteriors permetent que es pugui copiar un estructura de directoris d'una localització a un altre.

**NOTA:** el nom dels modes copy-out i copy-in pot ser una mica confús.



## ◆ Opcions

Opció	Abreviatura	Descripció
--reset-access-time	-A	Torna a establir el temps d'accés (atime) després de llegir el fitxer de forma qe no sembla que així estat llegit
--append	-A	Afegeix fitxers a un arxiu
--pattern-file=filename	-E filename	Utilitza els continguts del fitxer filename com a una llista de fitxers a extreure
--file=filename	-F filename	Utilitza el fitxer filename com a fitxer d'arxiu. Si no s'utilitza aquest paràmetre per defecte s'utilitza la entrada estàndard o la sortida estàndard.
--format=format	-H format	Utilitza un format específic per al fitxer d'arxiu. Els possibles formats són bin (per defecte, crea un fitxer binari), crc (nou format binari amb checksum), o <a href="#">tar</a>
No hi ha	-I filename	Utilitza el fitxer especificat en comptes de l'entrada estàndard. A diferència de -F no es redireccionen les dades de sortida.
--no-absolute-filenames	No hi ha	En mode copy-in (extreure) extreu els fitxers relatius al directriu de treball.
--unconditional	-u	Reemplaça els fitxers sense preguntar.
--list	-t	Mostra la taula de continguts de l'entrada
--verbose	-v	Mostra informació extra

## ◆ Exemples

- ◆ Combinant find, conducts i redireccions

```
$ find ./carpeta_a_arxivar | cpio -o > /media/KINGSTON/arxiu.cpio
```

- ◆ El fitxer resultat no es troba comprimit. Per comprimir podem utilitzar gzip:

```
$ find ./carpeta_a_arxivar | cpio -o | gzip > /media/KINGSTON/arxiu.cpio
```

## ◆ Extreure fitxers:

```
$ cpio -i < /media/KINGSTON/arxiu.cpio
```

```
$ gunzip -c /media/KINGSTON/arxiu.cpio | cpio -i
```

## ♦ dd (dataset definition)

- ♦ Ens permet arxivar tot un sistema de fitxers a baix nivell (les dades es copien byte a byte).
- ♦ També serveix per a fer conversions de dades crues (raw data). Les conversions es fan on-the-fly.
- ♦ Té una sintaxi una mica especial (derivada del Job Control Language - JCL – d'IBM):

```
$ dd if=/path/origen of=/path/destinacio [opcio=valor]...
```

- **if:** Input file
- **of:** output file

- ◆ **Podem realitzar, entre d'altres, accions com les següents:**
  - ◆ Emmagatzemar els continguts d'un CD/DVD, partició en un sol fitxer (una imatge de disc)
  - ◆ Fer una còpia de seguretat d'un disc dur i/o partició per tal de recuperar dades perdudes mitjançant informàtica forense.
  - ◆ Fer còpies de seguretat de regions específiques del disc com per exemple el sector d'arrancada o el MBR.
  - ◆ Llegir dades de dispositius especials del sistema com /dev/zero o /dev/random
  - ◆ Fer una còpia d'un disc o partició del qual no tenim drivers per accedir als seus continguts.
  - ◆ Restaurar còpies exactes de discs durs (eina de clonatge). Hi han solucions més avançades com clonezilla o fog

## ♦ Copiar per blocs. **bs** i **count**

- ♦ La mida del fitxer a transferir sempre és: `bs*count`
- ♦ Existeixen les següents mides de bloc:
  - ibs: bloc de lectura (input)
  - cbs: bloc de conversió
  - obs: bloc d'escriptura (output)
- ♦ Es poden utilitzar també els següents sufixos:
  - c: 1 byte, w: 2 bytes, b: 512 bytes (que és la mida habitual d'un sector de disc), kB: 1000 bytes, K: 1024 bytes, MB: 1000\*1000, M: 1024\*1024, GB: 1000\*1000\*1000, G: 1024\*1024\*1024 i successivament amb T, P, E, Z, Y.

## ♦ **Exemples**

- ♦ Crear una imatge ISO d'un CD/DVD

```
# dd if=/dev/cdrom of=/home/sergi/imatgeCD.iso bs=2048 conv=sync,notrunc
```

- ♦ Escriure tot un disc amb dades aleatòries (wipe)

```
# dd if=/dev/urandom of=/dev/sda
```

- ♦ Duplicar un disc dur

```
# dd if=/dev/sda2 of=/dev/sdb2 bs=4096 conv=notrunc,noerror
```

- ♦ Fer una imatge d'una partició

```
# dd if=/dev/sdb2 of=/home/sergi/partition.img bs=4096 conv=notrunc,noerror
```

# dd

- ♦ Fer una imatge d'una partició i enviar a una màquina remota

```
# dd if=/dev/sdb2 | ssh usuari@maquina "dd of=/home/usuari/partition.image"
```

- ♦ Duplicar el MBR

```
# dd if=/dev/sda of=/home/sergi/MBR.image bs=446 count=1
```

- ♦ Crear un fitxer buit d'una mida exacte

```
$ dd if=/dev/zero of=file1G.tmp bs=1G count=1
```

- ♦ Crear un disc d'arrancada USB

```
$ dd if=bin/gpxe.usb of=/dev/sdX
```

# dd i netcat

## ♦ Enviar una còpia de seguretat per xarxa

```
# dd if=/dev/sda1 bs=16065b | netcat < targethost-IP > 1234
```

### ♦ A la màquina remota

```
# netcat -l -p 1234 | dd of=/dev/sda1 bs=16065b
```



## ◆ Permet obtenir informació sobre un tipus de fitxer

- ◆ La sintaxi és:

```
$ file fitxer [fitxer2]...
```

- ◆ Identifica el tipus de fitxer pel seu contingut (no pas per l'extensió)
- ◆ Suporta fitxers binaris i fitxers de text

```
$ file protocol.png  
protocol.png: PNG image, 496 x 200, 8-bit/color RGB, non-interlaced
```

- Identifica la codificació. Si posa ASCII és que no té codificació

```
$ file /etc/passwd  
/etc/passwd: UTF-8 Unicode text  
$ file /etc/hosts  
/etc/hosts: ASCII English text
```

- Eines com iconv o geany ens permeten canviar codificacions



## 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>