

# Brevíssim manual d'ús de Linux

M. Vigo Anglada  
ETSEIB - UPC

## El sistema de fitxers

El sistema de fitxers és la forma que té el sistema operatiu d'organitzar la seva informació. Linux (i Unix) té un sistema de fitxers jeràrquic en forma d'arbre invertit, on cada node de l'arbre (branques) és un directori. Els directoris poden o no contenir fitxers.

## Directoris

Els directoris són estructures que contenen fitxers i altres directoris. Alguns dels directoris més importants en els sistemes Unix són els següents:

/ (arrel)	Directori arrel de l'arbre de directoris.
/usr/bin	Directori que conté programes executables.
/usr/local	Conté subdirectoris amb comandes per treballar localment
/usr/lib	Directori on hi ha les llibreries estàndard del sistema
/usr/include	director on hi ha els fitxers de capçaleres necessaris per compilar
/etc	Directori on hi ha implementades la majoria de comandes administratives
/tmp	Directori amb fitxers temporals
/home	Conté tots els directoris dels usuaris
/floppy	Directori on sol estar muntat el disquette

A tots els directoris hi ha dos directoris especials:

.	És el directori actual
..	És el directori superior (director pare)

La zona de treball d'un usuari està associada a un directori concret, el qual s'anomena directori arrel de l'usuari (*home directory*). Habitualment, sol dur el nom de l'usuari i estar situada dins del directori /home (per exemple, /home/azapatero).

## Pathnames

El pathname és una llista amb la jerarquia de noms de directoris que indica el recorregut per anar des d'un directori determinat a un altre. S'utilitzen per

especificar un lloc dins del sistema de fitxers. Hi ha dos tipus de pathnames: relatius i absoluts. Un pathname absolut indica el camí que cal seguir començant pel directori arrel, i per tant sempre comença pel símbol '/'. Un pathname relatiu indica un camí de directoris que cal seguir a partir del directori actual de treball (i mai no comença per '/').

En alguns shells (vegeu més avall) el símbol ~ serveix per indicar el nostre directori arrel i ~nomusuari indica el directori arrel de l'usuari que s'indica.

Exemples:

```
Pathanmes absoluts:  / /home /home/mgarcia/nou /usr/bin
                    /usr/include/X11/X11R6
Pathnames relatius:  . .. nou ../jmasos/inf2 ./nou .././usr/bin
                    ~/nou ~vferrer ~mserra/documents
```

## Comandes de manipulació de directoris

```
$ pwd          Informa del pathname absolut del directori actual
$ ls           Llista el contingut del directori actual
              Algunes opcions útils que es poden afegir són:
              ls -l, per mostrar tota la informació referent als
              fitxers i directoris (classe de fitxer, permisos, enllaços,
              propietari, mida en bytes, data de modificació, nom).
              ls -a, per mostrar també els fitxers ocults (els que comencen
              per '.', com per exemple .alias o .emacs)
              ls -R, per llistar el contingut de tots els subdirectoris
              (de forma recursiva).
$ cd           Canvia de directori. Cal indicar el pathname. Per exemple,
              cd Documents/Personals o bé cd /home/mserra
              Si no s'indica cap pathname, anirem al nostre directori arrel.
$ mkdir       Crea un directori. Cal indicar el pathname del directori
              que volem crear, el qual no pot existir prèviament.
$ rmdir       Esborra el directori del qual s'indica el pathname,
              el qual ha d'existir i estar buit.
```

## Fitxers

Un fitxer és una unitat d'informació que conté dades i que s'identifica per un nom. Els principals atributs d'un fitxer són els següents:

- Classe: d si és un directori, l si és un enllaç simbòlic a un altre fitxer, etc.
- Permisos: r indica permís de lectura, w d'escriptura i x d'execució. Hi ha a més a més tres nivells de permisos: propietari, grup i demés usuaris (*user*, *group*, *other*).
- Nombre d'enllaços físics al fitxer.
- Propietari.

- Mida en bytes.
- Data de darrera modificació.
- Nom del fitxer. L'identifica i ha de ser únic (dins del directori on està el fitxer).

Respecte al nom del fitxer, cal dir que en Unix/Linux el nom d'un fitxer és una cadena de caràcters qualsevol (tan llarga com vulguem), tot i que convé que el nom estigui compost només per lletres, números i alguns caràcters de puntuació (. \_ , -). Es distingeixen entre majúscules i minúscules (no és el mateix el fitxer `readme.txt` que el fitxer `ReadMe.txt`) i per convenció se sol posar un punt seguit d'una extensió al final del nom la qual indica el tipus de fitxer. Els fitxers tals que el seu nom comença pel caràcter '.' són fitxers ocults (no es llistaran a no ser que ho indiquem amb l'opció `-a` en fer `ls`).

## Comandes amb fitxers

\$ <code>cat fitxer</code>	Veure el contingut del fitxer.
\$ <code>more fitxer</code>	Veure el contingut del fitxer, però poc a poc. Mentre està aturat, si premem espai mostrarà la següent línia, si premem retorn la següent pàgina i si premem 'q' acabarà.
\$ <code>cp fitxer destí</code>	Fa una còpia d'un fitxer.
\$ <code>mv fitxer destí</code>	Mou de lloc o canvia el nom d'un fitxer.
\$ <code>rm fitxer [fitxer...]</code>	Eborra un (o més) fitxers. És interessant utilitzar sempre l'opció <code>-i</code> ( <code>rm -i fitxer</code> ), la qual ens demanarà confirmació, ja que un cop esborrat un fitxer no es pot recuperar (ATENCIÓ!).

## Propietaris, grups i permisos

Per a cada fitxer hi ha tres categories de permisos: usuari, grup i resta (*user*, *group*, *other*). L'usuari és el propietari del fitxer (habitualment, el qui l'ha creat), el grup fa referència als usuaris que pertanyen al mateix grup que el propietari i la resta es refereix a qualsevol altre usuari. El propietari d'un fitxer pot modificar el propietari del fitxer (l'usuari), amb la comanda `chown`, canviar el grup, amb `chgrp`, i modificar-ne els permisos, amb la comanda `chmod`. Descriurem el funcionament de la comanda `chmod` amb l'ajut d'uns quants exemples:

<code>chmod +r fitxer</code>	Activa el permís de lectura de tothom sobre el fitxer.
<code>chmod -x fitxer</code>	Desactiva el permís d'execució de tothom sobre el fitxer.
<code>chmod go-w fitxer</code>	Desactiva el permís d'escriptura sobre el fitxer per a la gent del grup i la resta d'usuaris.

Tots els directoris en el pathname absolut d'un fitxer han de tenir permís d'accés per tal que el fitxer sigui accessible. Per protegir un fitxer, és recomanable desactivar el permís d'escriptura del fitxer i també del directori on estigui.

Només el propietari del fitxer o el superusuari (administrador del sistema) poden modificar els atributs de possessió o del grup del fitxer.

## Altres comandes

\$ man <i>comanda</i>	Mostra el manual d'ús de la comanda especificada.
\$ man -k <i>paraula</i>	Mostra la llista de pàgines dels manuals on apareix la paraula buscada.
\$ ^C	Acaba l'execució del procés actiu
\$ ^Z	Suspèn (atura) l'execució del procés actiu
\$ bg	Deixa en execució en paral.lel un procés aturat (suspès)
\$ ps	Mostra la llista de processos actius
\$ kill -9 <i>idprocés</i>	Acaba l'execució del procés que s'indica

La majoria de comandes i aplicacions permeten l'opció -help (o de vegades -h o -help), per informar breument del seu funcionament i llisten les diferents opcions de què disposa la comanda.

## Caràcters especials i expressions regulars

En els sistemes Linux/Unix, com en qualsevol altre sistema operatiu, hi ha uns caràcters que tenen un sentit especial, a l'hora de fer matching amb els noms dels fitxers (de generar els noms dels fitxers). Aquests caràcters són:

- ? Es substituirà per un sol caràcter
- \* Es substituirà per zero o més caràcters qualssevol

Exemples:

\$ ls *.c	Llista tots els fitxers que acabin amb .c
\$ rm a*	Esborra tots els fitxers que comencin per a
\$ ls hola?.txt	Llista tots els fitxers que comencin per hola. segueixi un caràcter qualsevol i acabin amb .txt

També podem fer servir les classes de caràcters per fer la substitució. Per exemple: [aA] és la classe formada per a minúscula o per a majúscula; [a-z] és la classe formada per una lletra minúscula; [!t] és la classe formada per totes les lletres excepte la t; [!a-c] és la classe formada per totes les lletres menys la a, b i c. D'aquesta manera podem tenir les següents expressions: a?.\* és el conjunt de paraules que comencen per a, els segueix un caràcter qualsevol, un punt i qualsevol cosa; ab[1-5] és el conjunt de paraules que comencen per ab i els segueix un nombre entre 1 i 5; [a-zA-Z]1\* és el conjunt de paraules que comencen per una lletra (majúscula o minúscula) la segueix un 1 i després qualsevol cosa.

Atres caràcters especials són els següents:

< > S'utilitzen pel redireccionament d'entrada i sortida.  
 \$ S'utilitza per la substitució de variables.  
 ' S'utilitza per forçar l'execució d'una comanda.  
 | S'utilitza per a pipelines.

## Redireccionaments, pipelines i filtres

En els sistemes Unix/Linux podem redireccionar les entrades i les sortides, és a dir, obligar a què la informació que habitualment vindria des del teclat (el canal estàndard d'entrada) s'agafi d'un altre lloc (fitxer o dispositiu), o bé fer que el què habitualment sortiria per la pantalla (pel canal estàndard de sortida) vagi a parar en un altre lloc (fitxer o dispositiu). També podem lligar l'entrada d'una comanda amb la sortida d'un altre, utilitzant les pipelines.

Qualsevol comanda que llegeixi la entrada des de teclat, la podem redireccionar de manera agafi l'entrada des d'un fitxer. La forma de fer-ho és utilitzant el símbol <. Per exemple: \$ mail nom\_usuari < nom\_fitxer.

Per redireccionar la sortida d'una comanda cap a un fitxer, utilitzarem el símbol >. Per exemple, \$ ls -l > nom\_fitxer. Cal tenir en compte que en fer això el contingut el fitxer que recull l'entrada queda sobreescrit, i per tant perdrem el què contenia. Si volem que ens guardi la sortida generada al final d'un fitxer ja existent, podem fer: \$ comanda >> fitxer\_existent (per exemple, \$ man rand >> help.txt).

Les pipelines són útils si volem aprofitar la sortida d'una comanda com a entrada d'una altra. Sovint les utilitzem conjuntament amb els filtres per tal de facilitar-nos feina. El símbol associat és |, i la manera d'utilitzar-les és \$ comanda1 | comanda2. L'efecte és que la sortida de la comanda1 s'utilitza com a entrada de la comanda2. Per exemple:

```
$ ls | more      Llista els fitxers aturant la pantalla
$ ls | wc -l     Compta el nombre de fitxers del directori
$ ps -edalf | grep pepet
```

## Filtres

Un filtre és qualsevol comanda que llegeixi de teclat i generi la sortida per pantalla, per exemple: cat, more, sort, grep, wc.

```
$ cat fitxer    S'utilitza per a concatenar o visualitzar fitxers per la sortida
                estandard (pantalla).
$ sort [-ndtX] [+n camp] fitxer
                Ordena línies de camps. Les opcions signifiquen: -n Ordre
                numèric; -d Ordre alfabètic; -tX El separador dels camps és
                el caràcter X (per omisió és el tabulador); +n camp Indiquem
                el camp pel qual volem ordenar (començant a comptar des de 0).
```

\$ grep [-inv] plantilla fitxer

Cerca la plantilla dins del fitxer. Les opcions signifiquen:  
-i indica no fer diferència entre majúscules i minúscules;  
-v mostra les línies que no contenen la plantilla; -n mostra el número de línia. La plantilla sovint va entre comentos.

\$ wc [-lwc] fitxer

Compta les línies, paraules i caràcters d'un fitxer. Les opcions són: -l compta només línies; -w compta només paraules; -c compta només caràcters (Per omissió, compta les tres coses).

\$ find directori -name nom

Cerca un fitxer amb el nom que volem en el directori indicat.

Exemples:

\$ cat fitxer1 fitxer2 > fitxer3

\$ grep -i "llegir" fitxer.c

\$ sort -nt: +5 < /etc/passwd

\$ wc -l fitxer.txt

## Algunes adreces útils

<http://www.cecalc.ula.ve/bioinformatica/UNIX/>

<http://elqui.dsc.utfsm.cl/apuntes/labso/old/labso.1-Into-al-Linux/>

<http://www.imatica.org/articles/art2.html>