



## Il file system di Linux

Come Windows e praticamente in tutti i sistemi operativi, anche Linux organizza i files in una struttura ad albero

- In cima a tutto sta la directory (=cartella) radice, indicata con `"/`.
- Questa può contenere altre cartelle oppure files (=documenti). Un esempio è la cartella `home` che contiene i files degli utenti.
- Le sottocartelle si indicano quindi col nome preceduto da `/`. La sottocartella `home` si indicherà con `/home`
- Se esiste l'utente Topolino, la i suoi file sono nella sottocartella `Topolino` di `/home`, che viene indicata con `/home/Topolino`.

- Se qui c'è il file `Pluto.txt`, questo file è indicato in modo univoco come `/home/Topolino/Pluto.txt`. Dico che questo è il percorso assoluto del file.
- Se mi trovo nella cartella `/home`, in alternativa posso indicare anche il solo percorso di un file relativo alla mia posizione: il file precedente sarà allora `Topolino/Pluto.txt`
- Questa struttura si può ramificare a piacere, ed ogni cartella può contenere altre cartelle o files.
- La cartella dove mi trovo si può indicare con `."`, quella immediatamente superiore con `.."`

## Comandi principali

### Per gestire le directory (cartelle)

<b>mkdir</b>	crea una directory
<b>rmdir</b>	cancella una directory vuota
<b>cd</b>	cambia directory
<b>cd .</b>	non fa nulla
<b>cd ..</b>	passa nella directory superiore
<b>ls</b>	mostra i file e le sottodirectory
<b>pwd</b>	stampa a schermo la directory corrente

### Per gestire i file (documenti)

<b>touch</b>	crea o aggiorna la data di un file
<b>more e less</b>	visualizzano un file di testo un po' alla volta
<b>rm</b>	cancella un file
<b>cp</b>	copia uno o più file
<b>mv</b>	sposta uno o più file
<b>grep</b>	trova del testo all'interno di un file

### Per gestire i processi

<b>top</b>	vede i processi attivi e il consumo di CPU
<b>ps</b>	vede i processi dell'utente
<b>kill</b>	uccide un processo

## Uso e opzioni dei comandi più comuni

**cd** senza argomenti riporta nella home directory, quella in cui ci si trova dopo essersi collegati; per l'utente Topolino questa è /home/Topolino.

**cd ..** sposta nella directory superiore;

**cd dir** sposta nella directory di percorso relativo *dir*;

**cd /dir** sposta nella directory di percorso assoluto *dir*.

**rm** cancella un file solo se si ha il permesso di cancellarlo.

**rm -i** chiede conferma prima di cancellare.

**rm -f** cancella senza chiedere conferma.

**rm -rf** cancella ricorsivamente anche le sottocartelle senza chiedere conferma (**pericoloso!**)

**more** visualizza un file di testo un po' alla volta, ma senza la possibilità di tornare indietro

**less** è simile al precedente, ma permette di tornare indietro.

**cp file1 dir1** fa una copia del file1 nella directory dir1.

**cp file1 file2** fa una copia di file1 di nome file2 nella stessa directory.

- **Attenzione!!** file2 potrebbe già esistere e nessuno vi chiederebbe se volete proprio sovrascriverlo!
- se copiate più file in una directory mydir e poi vi accorgete che mydir in realtà non esiste, tutti i file saranno stati copiati in un file di nome mydir uno sull'altro: alla fine il file mydir conterrà solo una copia dell'ultimo file copiato;
- per evitare questo ed altri problemi usare l'opzione **cp -i** che chiede conferma prima di sovrascrivere.
- Per vedere cosa effettivamente state copiando provate **cp -v**.

**mv file1 mydir** sposta file1 nella directory mydir.

**mv file1 file2** rinomina file1 file2.

Valgono in questo caso i problemi segnalati per cp con l'aggravante che i file vengono anche cancellati dalla posizione originaria, e quindi se mydir non esiste potreste proprio perderli! Si usano le stesse opzioni che per **cp**.

## Caratteri jolly

Nei nomi di file \* indica una qualunque sequenza di caratteri: quindi `ls *.cpp` elencherà tutti i file il cui nome finisce per ".cpp". Invece `[aAx]` indica uno dei caratteri 'a', 'A' e 'x'. il nome del file `[bB]*.txt` indica tutti i file che cominciano per 'b' oppure 'B', finiscono per '.txt' e contengono in mezzo qualunque tipo e numero di caratteri.

Il punto interrogativo indica invece un unico carattere:

`ls ?a*` indica tutti i file e le cartelle che hanno una 'a' come seconda lettera

## Editor

Il programma deve essere scritto con un editor e non con un word processor. Buoni editor in ambiente carattere sono emacs e vi; in ambiente grafico gedit, l'editor integrato nell'ambiente di programmazione anjuta e l'editor per programmatori geany. È importante imparare a usare subito i movimenti del cursore, pagina su e giù, e il “taglia, copia e incolla” che di solito ha degli shortcut con CTRL-C per copia, CTRL-X per taglia e CTRL-V per incolla.

Come esempio, considero il programma [gedit](#), ma molti altri programmi hanno comandi simili

- Lanciare il programma "gedit".
- dal menu scegliere "File" e poi "Open".
- Comparirà una finestra in cui si può scegliere il file da editare: selezionarlo e scegliere "Open".
- Ci si può muovere nel file con le frecce oppure con il mouse.

- Per selezionare del testo si può muovere le frecce tenendo premuto il tasto "Shift" ("Maiuscole") oppure portare il mouse all'inizio della parte da selezionare, poi spostarlo alla fine tenendo premuto il tasto sinistro.
- A questo punto si preme il tasto destro del mouse e comparirà un menu dal quale si può scegliere se tagliare, copiare o cancellare; le prime due operazioni si possono fare anche con CTRL+X e CTRL+C.
- Fatto questo, si può inserire il testo selezionato in un'altra posizione nel file spostandosi là e scegliendo "Incolla" dal menu del tasto destro del mouse oppure premendo CTRL+V.
- si può cercare del testo nel file con CTRL+F o rimpiazzando del testo con altro testo con CTRL+H.
- Finite le modifiche del file, lo si può salvare scegliendo "File" e quindi "Salva".

## Documentazione in Linux

Ci sono varie utility per avere informazioni sui singoli comandi Linux e sulle loro possibili opzioni.

**man** Si può sapere tutto su un comando del sistema operativo linux con il comando **man**. Ad esempio **man ls** mostra tutte le opzioni per avere il listato di una cartella e **man grep** mostra le molte possibilità del comando grep.

**info** Sono un tentativo del progetto GNU di scrivere una documentazione in forma ipertestuale in tempi in cui internet e il linguaggio HTML non esistevano. Sono piuttosto complicate da navigare, ma per fortuna oggi è possibile visualizzarle graficamente. Il modo testuale inizia comunque col comando

*info comando*

**yelp** Questo, come altri comandi simili, è un'interfaccia navigabile dei programmi di documentazione. È navigabile, e permette di muoversi tra opzioni e comandi cliccando il mouse.

**Apropos** Il comando `apropos` serve quando vogliamo fare qualcosa ma non sappiamo come farlo. Se avete sonno provate *apropos sleep* oppure *apropos game* se non sapete cosa fare.

**cartella /usr/share/doc** In questa cartella molti programmi installano i file con la documentazione. Darci un'occhiata, se non si è soddisfatti delle altre informazioni, può riservare piacevoli sorprese.

**Internet** Ovviamente questa è la fonte di documentazione più completa, se non altro perché comprende tutte le precedenti. Una possibilità è andare a vedere il sito [www.cplusplus.com](http://www.cplusplus.com) per il linguaggio  $C_{++}$  e il sito [www.nr.com](http://www.nr.com) per gli algoritmi numerici.