

Alessandro Vicini

Università di Milano

- **Comandi base in linux + LaTeX**
- **Mathematica: funzioni di base**
- **Mathematica: prime applicazioni**

Sistema operativo + File-system

Il Sistema operativo:

- gestisce il funzionamento del computer
- controlla il funzionamento delle periferiche e dei devices
- controlla il flusso dei dati dalle memorie al processore e viceversa
- permette all'utente di eseguire programmi

Le singole operazioni/funzioni svolte dal S.O. sono dette processi

La memoria del computer si divide in:

- memorie fisse (disco rigido, floppy, cdrom)
- memorie di sistema (RAM) di supporto al processore

I dati e programmi personali di un utente risiedono nelle memorie fisse sotto forma di files

Il file-system di un computer è il modo con cui i files sono strutturati, con cui vengono scritti, letti, modificati e cancellati

Un file è una porzione di memoria caratterizzato da un nome e dalla sua dimensione (è costituito infatti da un certo numero di bytes).

Il file-system in linux ha una struttura ad albero.

Il punto di partenza, la radice, *root*, è indicato da /

Ogni ramo, *directory*, ha un nome

p.es.: /home/vicini

I files possono essere considerati come le foglie dell'albero

Un file è univocamente identificato dalla sua posizione all'interno dell'albero

ovvero dal suo nome completo

p.es.: /home/vicini/prova.txt

E' possibile muoversi lungo l'albero utilizzando il comando `cd`

- per scendere: `cd nomedir`
- per risalire: `cd ..`

Help

Ogni comando ha un minimo help on-line che può essere ottenuto digitando

```
nomecomando -help
```

Maggiori informazioni si possono ottenere consultando la pagina on-line del manuale, digitando

```
man nomecomando
```

Nel caso non conosciate il nome esatto di un comando, potete eseguire

```
man -k parolachiave
```

Comandi base di gestione dei files

E' possibile creare/cancellare directories con i comandi `mkdir / rmdir`

- per creare: `mkdir nomedir`
- per cancellare: `rmdir nomedir`

Ogni directory può contenere files e/o i nomi di ulteriori sub-directories

E' possibile creare un file utilizzando un editor di testo come p.es. `emacs, vi, pico`

E' possibile elencare i files presenti in una directory con il comando `ls`

In linux spesso ad un comando è possibile aggiungere delle opzioni
`ls -l` *informazioni estese su ciascun file*
`ls -a` *elenca anche i files .nomefile presenti nella directory*

Il simbolo * rappresenta una generica stringa di lunghezza arbitraria.
P.es. `ls *.txt` *elenca tutti i files che terminano con .txt*

Il significato dei caratteri di un file può essere vario:

- un testo letterario (appunti, tesi,...)
- un programma in un dato linguaggio (comprensibile p.es. da un compilatore)
- un programma in linguaggio macchina, eseguibile dal processore

E' possibile determinare il tipo di contenuto del file utilizzando il comando

`file filename` Al contrario di Windows, il tipo di un file NON è determinato dalla sua estensione. Ovviamente l'estensione risulta un pratico aiuto per l'utente e per molte applicazioni.

Un file è caratterizzato dai suoi permessi
ci sono 3 categorie di accessi: world, gruppo, utente
per ciascuna categoria è possibile indicare la possibilità di leggere (r) scrivere (w) eseguire (x) il file

E' possibile modificare questi permessi utilizzando il comando

`chmod [ugoa...]+-=[rwx...] filename`

p.es.: `chmod u+x prova.txt`

E' possibile copiare un file con il comando

`cp oldname newname`

fornendo l'estensione completa è possibile spostare un file da una directory a un'altra

E' possibile spostare un file con il comando

`mv oldname newname`

Differenza: cp conserva oldfile e ne produce una seconda copia, mentre mv elimina oldname e assegna al file il nuovo nome (eventualmente la nuova collocazione)

E' possibile cancellare un file con il comando

`rm filename`

per avere maggiore sicurezza è possibile utilizzare

`rm -i filename`

che richiede conferma prima di cancellare il file

Altri comandi di gestione dei files

Il comando di base per la stampa di un file è:

```
lpr -Pprintername filename
```

Utili opzioni sono la stampa fronte-retro (-K2) e la stampa di due pagine per facciata (-N2). Questi dettagli dipendono dalla configurazione locale.

La sola lettura di un file (senza scrittura) è possibile con i comandi

```
less filename
```

```
more filename
```

è possibile effettuare ricerche di regular expressions

Quando si hanno a disposizione diversi set di risultati è spesso utile unirli in un solo file, utilizzando p.es. il comando

```
cat file1 file2 > file3
```

Il simbolo > indica che l'output dell'operazione cat file1 file2 viene indirizzato in file3

Spesso la dimensione dei file grafici, Postscript,... richiede una loro compressione, p.es. per spedirli per e-mail più agevolmente.

Per lo stesso motivo i files scaricabili dalla rete sono compressi, per accelerare il download.

I comandi utili per questo tipo di operazioni sono:

```
gzip, gunzip, tar
```

Il comando

```
gzip filename           produce  filename.gz
```

```
gunzip filename.gz     produce  filename
```

Un programma come gv visualizza direttamente anche files gzippati

Il comando tar serve per raccogliere in un archivio (ovvero in un unico file maneggiabile come un singolo oggetto) svariati files o addirittura files e sottodirectories, di cui viene preservata la struttura.

Le opzioni più comunemente applicate sono:

<code>tar -xvf filename.tar</code>	estrae dall'archivio il suo contenuto
<code>tar -cvz *.* > filename.tar</code>	crea l'archivio filename con tutti i files della directory corrente utilizzando la compressione gzip

Gli mtools

La gestione del floppy disk può essere resa simile al linguaggio DOS tramite una serie di comandi noti come mtools.

I più comuni sono:

`mdir`

`mcopy fileiniz filedestinaz`

Il loro utilizzo dipende dal tipo di installazione di linux e dalla macchina

Risulta spesso necessario, per poter utilizzare un lettore cdrom, montare il device.

I comandi rilevanti sono mount, umount

Il filesystem di linux richiede infatti che al device fisico che contiene il CD (nella fattispecie il lettore) sia associato un nome "logico", che può essere manipolato (p.es. copiato). Al termine di tutte le operazioni è necessario smontare il device (interrompere il link) per recuperare il proprio CD.

Altri comandi utili

I processi di un utente sono visualizzati tramite il comando `ps`

Ogni processo è identificato univocamente da un numero, la sua PID (process identity). In casi disperati è possibile bloccare un processo con il comando `kill -9 PID`

L'utilizzo delle risorse di una macchina è visualizzabile tramite il comando `top`

E' FONDAMENTALE che ciascun utente acquisti piena consapevolezza delle risorse (memoria e CPU) consumate dai suoi programmi.

Per la ricerca di una parola all'interno di uno o più files è utile il comando

`grep`

p.es. `grep pippo *.txt`

Un concetto fondamentale di Unix è la possibilità di indirizzare files come input di nuove applicazioni (filtri) in maniera elementare

Il simbolo the pipe `|` permette di realizzare queste connessioni tra comandi/programmi diversi.

Considerate p.es. l'istruzione `ps aux | grep bash`

Due files molto simili possono essere confrontati con `diff file1 file2`

Una stringa può essere ricercata nel filesystem (ovvero tra tutti i nomi di files) con il comando

`locate stringa`

Il nome completo di un file eseguibile, se compreso nel path della shell, può essere visualizzato con il comando `which`

Un file eseguibile può essere eseguito digitando `./filename`

La shell

L'utente dialoga con il computer digitando istruzioni sulla linea di comando. La finestra e il prompt che compaiono all'utente visualizzano la cosiddetta shell.

Si tratta di un processo che il sistema operativo lancia, dando la possibilità all'utente di digitare un certo numero di comandi. Esistono diverse shell che si differenziano per i comandi disponibili. Oltre ai comandi visti finora ne esistono altri contenuti in `/usr/bin`

e contenuti in tutte le directories incluse nel path.

Il path può essere visualizzato con `echo $PATH` L'utente può personalizzare la sua shell editando nella home directory un file `.bashrc` in cui aggiunge al path di default anche altre directories `PATH = $PATH:/nomedirectory` Può inoltre rendere default alcune opzioni introducendo degli alias
`alias 'rm=rm -i'`

L'utente può scrivere degli script, ovvero delle sequenze di comandi di shell che possono essere richiamati tramite un singolo file, reso eseguibile.

Connessione con l'esterno

Il modo sicuro per collegarsi ad una macchina remota è

```
ssh utente@macchina.dominio
```

Il modo sicuro per copiare files da/su una macchina remota è

```
scp fileiniz macchina.dominio:directory/filedestinaz
```

Redazione di documenti

E' uso scrivere testi scientifici utilizzando il linguaggio LaTeX.

Il file viene compilato (file.tex → file.dvi) e successivamente trasformato in un file Postscript (file.dvi → file.ps).

I comandi sono `latex nomefile.tex ; dvips nomefile.dvi`

Per visualizzare il risultato si usa il programma `gv`

La conversione al formato PDF è immediata

```
ps2pdf file.ps
```

ciao