

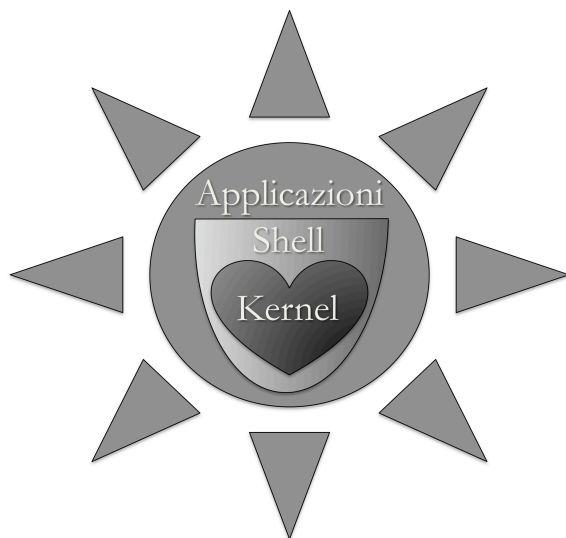
14. Introduzione a Linux

Linux è il sistema operativo derivato da **Unix**, uno dei più famosi sistemi operativi al mondo, sviluppatosi verso la metà degli anni Settanta del secolo scorso come sistema *multitasking* e *multiuser*; è quindi sistema operativo che permette a diverse persone di lavorare su uno stesso computer in parallelo, lanciando diverse applicazioni contemporaneamente. Ideato nel 1991 dall'allora studente Linus Benedict Torvalds, Linux è cresciuto negli anni grazie soprattutto alla collaborazione volontaria di programmatori Unix ed esperti di programmazione che navigavano in Internet, sino a divenire un sistema operativo robusto, efficiente e versatile, mantenendo in molte delle sue distribuzioni la filosofia *free and open-source*: Linux è gratuito e liberamente modificabile grazie alla condivisione del codice sorgente. Moltissime delle applicazioni in ambito bioinformatico nascono e sono disponibili su sistemi operativi Unix/Linux.

Questa introduzione a Linux, certamente incompleta e limitata, ha lo scopo di descriverne la struttura e i comandi più comunemente utilizzati. Per approfondire la conoscenza di questo sistema operativo, si rimanda ai numerosi testi disponibili sull'argomento, anche in rete (si veda per esempio la documentazione disponibile su www.linux.it).

Architettura di Linux

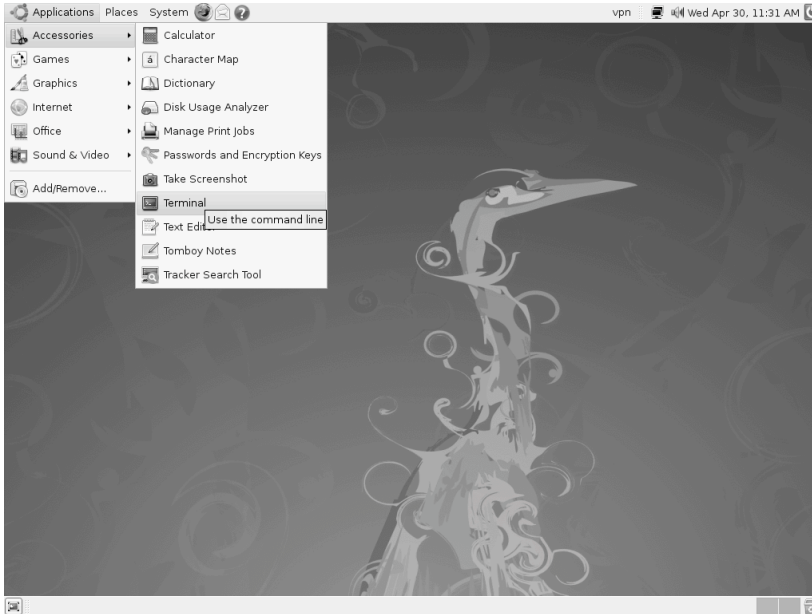
Come Unix, Linux è l'insieme di tre componenti: kernel, shell e applicazioni.



Il kernel costituisce il nucleo del sistema operativo. Si tratta di un software che ha il compito di fornire ai programmi in esecuzione sull'elaboratore un accesso sicuro e controllato all'hardware. Dato che può essere eseguito simultaneamente più di un processo, il kernel ha anche la responsabilità di assegnare una porzione di tempo-macchina e di accesso all'hardware a ciascun programma. La shell è il programma che permette agli utenti di comunicare con il kernel e di avviare i programmi. Una shell testuale è un programma con interfaccia a linea di comando:

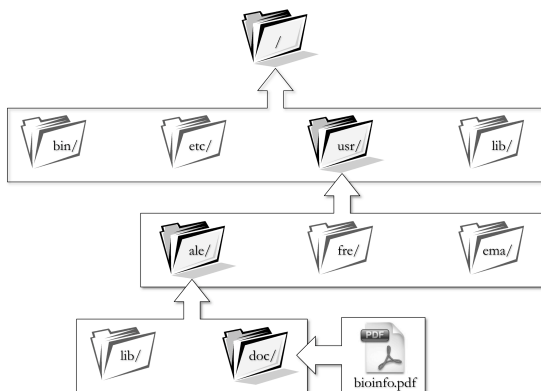
l'utente digita un comando, cioè richiede l'esecuzione di un programma, e il programma eseguito può interagire con l'utente ed eseguire il compito richiesto.

Su una delle distribuzioni di Linux più diffuse, **Ubuntu**, la shell può essere aperta selezionando *Applicazioni -> Accessori -> Terminale* in alto a sinistra del desktop, come mostrato in figura:



Le applicazioni costituiscono l'insieme dei programmi di cui dispone il sistema operativo. L'ambiente grafico di un sistema operativo rappresenta un esempio di applicazione. Quando le applicazioni vengono lanciate dall'utente o dal sistema operativo e diventano attive in memoria, sono chiamate **processi** e sono associate a un **PID** (*process identifier*).

I processi utilizzano solitamente dei file, collezioni di dati conservati in una memoria permanente. Esempi di file possono essere documenti o programmi in formato di testo (**ASCII**) e dati o programmi in formato binario. I file sono raggruppati in directory (cartelle). L'organizzazione dei file negli ambienti Linux/Unix (*filesystem*) ha una struttura gerarchica. Il primo elemento della gerarchia è chiamato **radice** (*root*) ed è indicato con il simbolo **"/"**. Ogni file viene quindi identificato univocamente da un percorso (*pathname*). Per esempio, il *pathname* completo del file *bioinfo.pdf* nella seguente figura:



è */usr/ale/doc/bioinfo.pdf*.

Oltre al *pathname* completo, esiste un *pathname* relativo alla directory nella quale ci troviamo. Per esempio, se ci trovassimo nella directory “doc”, il *pathname* relativo di *bioinfo.pdf* sarebbe:

./bioinfo.pdf (dove il punto “.” è l’alias della directory nella quale ci troviamo).

Se ci trovassimo nella directory “ale”, il *pathname* relativo di *bioinfo.pdf* sarebbe:

./doc/bioinfo.pdf

Al termine di questo documento è fornita una breve descrizione del contenuto delle *directory* comunemente presenti in un *filesystem* Unix/Linux.

Primi passi con la shell Unix

Aprirete una shell come descritto nel paragrafo precedente. Sulla linea di comando (*prompt*) troverete solitamente il *pathname* della vostra *directory* di partenza (chiamata **home**) e il cursore in attesa di ricevere comandi. Prima di digitare qualsiasi comando, è necessario approfondire la nostra discussione sulla logica dei comandi Linux/Unix.

Una frase in linguaggio naturale come «*dammi subito un fazzoletto gentilmente*» in Linux/Unix dovrebbe essere formulata in questo modo:

dammi -subito -gentilmente fazzoletto

dove:

dammi è il **comando**;

-gentilmente -subito sono le **opzioni**;

fazzoletto è il **parametro**.

Una «frase» siffatta in Linux/Unix sarebbe sinteticamente formulata come:

dmm -g -s fazzoletto

Opzioni e parametri sono chiamati **argomenti**.

Notate che lo spazio è sempre il separatore tra comando e argomenti e tra i vari argomenti, i nomi di comandi, opzioni e parametri *non* ammettono spazi interni e per eseguire un comando occorre usare il tasto *Invio* (*Enter*).

Inoltre, bisogna prestare attenzione al fatto che Unix e Linux fanno distinzione tra maiuscole e minuscole: *cd* (il comando necessario a cambiare la directory corrente) e *CD* non sono equivalenti.

In Unix e Linux *miofile.txt*, *MioFile.txt* e *MIOFILE.txt* sono tre file diversi, ma se vengono trasferiti in Windows diventano lo stesso file. Windows non distingue, infatti, tra maiuscole e minuscole.

- Per visualizzare l’elenco dei file contenuti della directory, digitate **ls** (**list**) (+ *Invio*).

Di seguito utilizzeremo le seguenti convenzioni: i caratteri scritti in **grassetto e sottolineato** sono le istruzioni che devono essere digitate nel computer così come sono scritte in questa esercitazione, mentre i caratteri scritti in corsivo rappresentano nomi generici di file o directory.

- Digitate **pwd** (*print working directory*) per visualizzare il *pathname* della directory corrente.

Il risultato dovrebbe essere di questo tipo: */home/home_directory*.

- Create una directory con il comando: **mkdir** *nome_directory*
- Entrate nella directory appena creata con: **cd** *nome_directory*
- digitate **ls** : la directory appena creata è vuota.
- Ora tornate alla directory precedente con il comando: **cd ..** (“..” è un alias che indica la *directory* precedente).

- Digitate ora il comando **ls** seguito dall'opzione **-l** (lettera "l" minuscola): **ls -l**.

Otterrete un risultato simile al seguente:

```

sh-3.2$ ls
CLC_Data  Documents  Icon?      Library    Logs       Music      Public
Desktop   Downloads  Incomplete LimeWire   Movies     Pictures   Sites
sh-3.2$ ls -l
total 120
drwxr-xr-x  6 alessandro wheel  204  2 Lug 18:46 CLC_Data
drwxr-xr-x+ 5 alessandro wheel  170 18 Nov 17:28 Desktop
drwxr-xr-x+ 43 alessandro wheel 1462 29 Set 11:18 Documents
drwx---r-x+ 56 alessandro wheel 1904 10 Nov 17:24 Downloads
-rw-r--r--@ 1 alessandro wheel   0 15 Giu 10:48 Icon?
drwxr-xr-x  7 alessandro wheel  238  4 Ago 15:44 Incomplete
drwxr-xr-x+ 53 alessandro wheel 1802  7 Ott 18:46 Library
drwxr-xr-x  3 alessandro wheel  102 22 Ott 2008 LimeWire
drwxr-xr-x  6 alessandro wheel  204 14 Mag 2009 Logs
drwxr-xr-x+ 11 alessandro wheel  374  4 Set 20:39 Movies
drwxr-xr-x+  9 alessandro wheel  306  3 Lug 2008 Music
drwxr-xr-x+ 27 alessandro wheel  918 26 Ago 11:53 Pictures
drwxr-xr-x+  6 alessandro wheel  204 12 Nov 2008 Public
drwxr-xr-x+  5 alessandro wheel  170 29 Mar 2007 Sites
sh-3.2$ █

```

Nella lista sarà presente anche la directory da voi appena creata.

Nell'elenco sono specificate le seguenti informazioni (da sinistra a destra):

1. il tipo di file ("-" file normale, "d" directory, "l" link);
2. i permessi (vedi dopo);
3. il proprietario del file;
4. il gruppo di appartenenza del proprietario del file;
5. la grandezza del file in byte;
6. la data di ultima modifica;
7. il nome del file o della directory.

I permessi

Come già accennato, Linux è un sistema multiutente. Per ogni file ci sono 4 categorie di utenti: **root** (l'amministratore del sistema), **owner** (il proprietario del file), **group** (il gruppo di utenti), **world** (il «resto del mondo»). Ogni utente può leggere, modificare ed eseguire (nel caso di programmi) i file in base ai permessi che ha su di esso.

Il blocco di caratteri «rw-r--r--» (o una qualsiasi altra combinazione di **r**, **x**, **w** o **-**) rappresenta i permessi di accesso ai file. I primi 3 (rw-) sono riferiti all'**owner**. Il secondo blocco di 3 caratteri (r-) è riferito al **group** e l'ultimo blocco (r--) è riferito alla categoria **world**.

La prima posizione di ogni blocco rappresenta il permesso di lettura (**r**) del file, la seconda il permesso di scrittura (**w**) sul file e la terza il permesso di esecuzione (**x**) del file (se si tratta per esempio di un programma). Un trattino (-) in una qualsiasi posizione indica l'assenza del permesso corrispondente.

Nota bene: per accedere a una directory, bisogna avere il permesso di esecuzione su di essa.

Creazione di un file e modifica dei permessi

In Linux non esiste un vero e proprio comando per creare un file (anche se esistono alcuni comandi, per esempio **touch**, che creano un file se questo non esiste). Si può invece aprire un editor di testo e, dopo aver inserito il contenuto desiderato, salvarlo (ci sono molti editor disponibili, il più famoso

dei quali è forse **vi**; a causa della complessità di *vi*, però, in questa esercitazione consigliamo di usare un editor di testo alternativo, come **kate**, simile al diffusissimo Word di Microsoft, o **gedit**, un altro editor assai diffuso).

- Collegatevi al sito della Protein Data Bank (www.rcsb.org) e visualizzate le coordinate della proteina con codice 1BJ4 (SHMT umana);
- Copiate le coordinate e salvatele in un file di testo;
- Aprite il file e rimuovete le righe contrassegnate dal campo *REMARK*;
- Salvate il file.

Modificheremo ora i permessi del file appena creato. Il comando da utilizzare è **chmod**:

chmod *opzioni nome_file*

chmod è seguito da tre numeri (*opzioni*) che modificano i permessi per il proprietario del file, per il gruppo e per il resto del mondo, rispettivamente.

Questi numeri sono:

- 4 rende il *file* disponibile in sola lettura
- 2 rende il *file* disponibile in sola scrittura
- 1 rende il *file* disponibile in sola esecuzione

e qualsiasi combinazione di essi: per esempio, 7 (4+2+1) rende il file disponibile in lettura, scrittura ed esecuzione; **chmod 755 nome_file** rende *nome_file* disponibile in lettura, scrittura, esecuzione per l'utente, lettura ed esecuzione per il gruppo, lettura ed esecuzione per il resto del mondo.

Provate a rendere il file che avete creato disponibile in lettura, scrittura ed esecuzione per il proprietario, lettura ed esecuzione per il gruppo e lettura ed esecuzione per il resto del mondo. Potete aiutarvi con il manuale disponibile per qualsiasi comando Linux: è sufficiente digitare: **man nome_del_comando**. L'opzione **-k** usata con il comando **man** e seguita da un qualsiasi termine permette di ricavare informazioni sui comandi relativi al termine stesso. Per esempio:

man -k screen

visualizzerà tutti i comandi correlati allo schermo. Per uscire dal manuale digitate **:q**

Altri comandi utili

Fin ora abbiamo mostrato alcuni comandi utili per muoversi nelle directory, creare file e modificarne i permessi. Il numero di comandi disponibili in Unix/Linux è altissimo; di seguito riportiamo alcuni dei comandi utilizzati per le operazioni più comuni.

Copiare un file

cp *fn1 fn2*

dove **cp** è il comando, *fn1* il nome di un file esistente e *fn2* il nome del *nuovo file*. È necessario specificare il *pathname* dei file se non appartengono alla directory corrente. Provate a copiare il file creato precedentemente.

Cancellare un file

rm *fn*

dove **rm** è il comando e *fn* è il nome del *file*. Cancellate la copia del *file* generato precedentemente.

Cancellare una directory

rmdir *dirn*

dove **rmdir** è il comando e *dirn* è il nome della directory. La directory deve essere vuota. Cancellate la directory creata precedentemente.

Spostare o rinominare un *file*

mv *fn1 fn2*

dove **mv** è il comando, *fn1* è il nome del *file* e *fn2* è il nuovo nome. Rinominate il *file* precedentemente creato.

Visualizzare i processi in esecuzione

top

per uscire da top premere la combinazione di tasti Ctrl-C

Domande

- 1 In Linux esistono dei file nascosti nelle directory, caratterizzati dalla presenza di un punto prima del nome del file (p.e. *.netscape*). Qual è l'opzione da dare a **ls** per visualizzare questi file (aiutatevi con il comando **man**)?
- 2 Qual è l'opzione da dare a **rm** per rimuovere i file di una directory e di tutte le directory contenute in essa, in maniera ricorsiva?
- 3 Qual è il comando per rendere un file accessibile in lettura e scrittura al proprietario e al gruppo, e in lettura al resto del mondo?

Descrizione del contenuto delle directory comunemente presenti in un filesystem Unix/Linux

/ Directory di partenza del sistema

/bin *Contiene comandi importanti*

/dev *Contiene file di accesso ai dispositivi fisici del calcolatore*

/lib *Contiene le librerie necessarie al funzionamento dei programmi*

/etc *Contiene file per l'amministrazione del sistema*

/tmp *Contiene file temporanei*

/var *Contiene file che tendono a crescere di dimensioni*

/home *Contiene le directory degli utenti*

/sbin *Contiene i programmi di partenza del sistema*

/usr *Contiene generalmente la maggior parte del sistema operativo*