

Il Sistema Linux

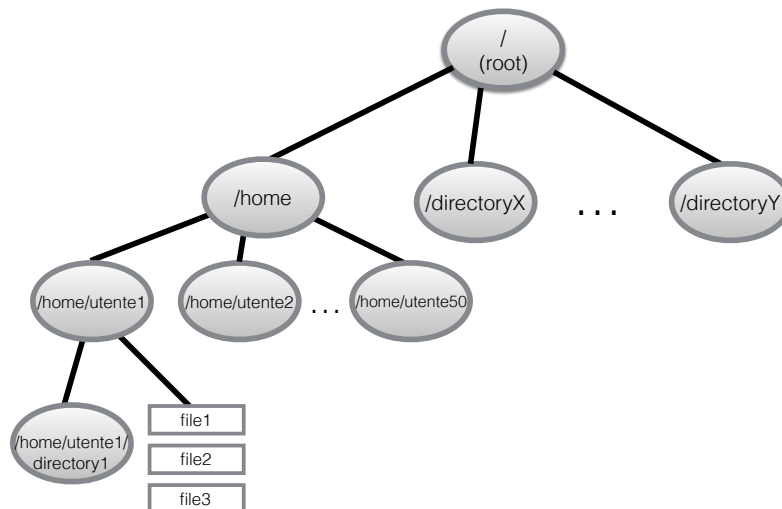
► Procedura di Login

Una volta acceso il computer e caricato il sistema operativo, l'utente esegue la procedura di Login. Il Login richiede uno username e una password. Per ciascun gruppo sono stati preparati degli account i cui nomi sono lcYYXX dove YY sono le iniziali del docente e XX=0, 1, ..., 50

```
1 Username: lcng07
2 Password: lcng07
```

► Cartelle e documenti

Quando l'utente entra nel sistema gli viene assegnata un'area di disco dedicata ai propri documenti (files) che può essere divisa in cartelle (directories). Una directory rappresenta una divisione logica dell'area di disco. Le directories hanno un nome che segue le stesse convenzioni dei nomi dei files. Una directory può contenere altre directories in un albero gerarchico. Il livello più alto della gerarchia è rappresentato dalla directory *radice* (root), rappresentata dal simbolo / (slash). La radice, contiene altre directories tra cui la directory *home* che contiene a sua volta le directories dei singoli utenti. L'albero delle directories si può navigare separando i nomi delle directories con uno il simbolo /. Il percorso di una directory descrive la sua collocazione nella struttura gerarchica. Ad esempio, se la cartella /home/ contiene la sottocartella utente1 (come nell'esempio grafico), allora il percorso della directory sarà /home/utente1/. La stessa cosa vale se all'interno della directory ci sia un file: ad esempio, il percorso del file chiamato file1 è /home/utente1/file1. Si noti come le directories possano contenere sia files che altre directories (vedere Figura 1).



► La shell

In Linux (e in UNIX) si interagisce con il sistema attraverso un programma chiamato shell che viene invocato automaticamente al Login. I comandi sono dati sotto la shell che li interpreta, esegue e scrive il risultato su terminale. La linea comandi è caratterizzata un prompt (\$) e da un cursore (che generalmente è rappresentato dal carattere _ o da un rettangolino lampeggiante). Nel prompt possono essere indicate alcune informazioni come ad esempio il nome dell'utente

```
1 utente1:~$
```

► Visualizzazione della directory corrente (pwd)

Il comando `pwd` (*print working directory*) visualizza la posizione attuale dell'utente nella gerarchia nell'area del disco assegnata

```
1 utente1:~$ pwd
2 /home/utente1
```

► Visualizzazione dei files e directories nella directory corrente (ls)

Il comando `ls` (*list*) restituisce l'elenco dei files e delle cartelle della directory corrente.

```
1 utente1:~$ pwd
2 /home/utente1
3 utente1:~$ ls
4 directory1  file2
5 file1      file3
6 utente1:~$
```

Alla fine dell'elenco riappare il prompt, dove si può digitare un altro comando. Alcune opzioni utili per il comando `ls` sono:

1. `ls -a` mostra anche i file nascosti
2. `ls -l` mostra tutti i dettagli di ogni file

Se al comando `ls` si passa il nome di un file o di una directory il comando mostra solo il file o la directory indicati. Il nome si può esprimere usando la wildcard *: un carattere speciale che corrisponde a tutti i possibili caratteri. Per esempio

```
1 utente1:~$ ls -l *.pdf
```

mostra, con tutti i dettagli, la lista dei files che terminano con `.pdf`. In generale tutti i comandi Unix si compongono di una parola chiave (`ls`) cui seguono zero o più parametri (`*.pdf`) e/o dei modificatori (`-l`) che ne alterano il comportamento di default. Per conoscere tutti i possibili usi di un comando si può usare il comando `man` (manuale)

```
1 utente1:~$ man ls
```

► Creare una nuova directory (mkdir)

Per creare una directory, nella shell si usa il comando mkdir (make directory)

```
1 utente1:~$ mkdir directory1
```

dove, in questo esempio, directory1 è il nome della cartella che abbiamo creato

► Spostamento in un'altra directory (cd)

Il comando cd (change directory) seguito dal percorso di una directory permette di spostarsi all'interno di essa

```
1 utente1:~$ cd /home/utente1/directory1
2 utente1:~$ pwd
3 /home/utente1/directory1
```

usando il comando cd .. si sale di un livello nella gerarchia di cartelle. Per tornare nella directory principale (/home/utente1) si può dare il comando cd ~. infatti il simbolo ~ rappresenta la propria home directory.

```
1 utente1:~$ pwd
2 /home/utente1/directory1
3 utente1:~$ mkdir my_dir
4 utente1:~$ cd my_dir
5 utente1:~$ pwd
6 /home/utente1/directory1/my_dir
7 utente1:~$ cd ~
8 utente1:~$ pwd
9 /home/utente1
```

Nell'esempio precedente si poteva tornare da mydir alla home directory anche eseguendo due volte il comando cd .. come segue

```
1 utente1:~$ pwd
2 /home/utente1/directory1/my_dir
3 utente1:~$ cd ../../
4 utente1:~$ pwd
5 /home/utente1
```

I due punti .. rappresentano dunque la directory che si trova al livello superiore di quella attuale. Il punto singolo invece rappresenta la directory corrente.

► Spostare e cambiare nome a files e directories (mv)

Per cambiare nome ad un file, si esegue un comando che "sposta" il file originale in un altro di nome diverso. Il comando da utilizzare è mv (move):

```
1 utente1:~$ pwd
2 /home/utente1
3 utente1:~$ ls
4 directory1  file2
5 file1       file3
6 utente1:~$ mv file1 nuovo_file
7 utente1:~$ ls
8 utente1:~$ directory1  file3
9         file2          nuovo_file
```

Lo stesso comando può essere utilizzato per cambiare nome ad una cartella o spostarla in un'altra cartella

```
1 utente1:~$ ls
2 directory1  file2
3 file1       file3
4 utente1:~$ mkdir my_dir
5 utente1:~$ ls
6 utente1:~$ directory1  file1  file3
7         my_dir         file2
8 utente1:~$ mv my_dir directory1/
9 utente1:~$ cd directory1/
```

Nell'esempio precedente è stata creata una cartella my_dir e spostata nella directory1.

```
1 utente1:~$ utente1:~$ pwd
2 /home/utente1/directory1/
3 utente1:~$ ls
4 my_dir
```

► Cancellare un file o una directory (rm)

Per cancellare un file si usa il comando rm (remove)

```
1 utente1:~$ ls
2 directory1  file2
3 file1       file3
4 utente1:~$ rm file1
5 utente1:~$ ls
6 directory1  file3
```

Analogamente, con lo stesso comando è possibile cancellare una directory vuota. Per cancellare una directory e tutto il suo contenuto (pensateci bene prima di usare questo comando!) si usa l'opzione -r

```
1 utente1:~$ rm -r directory1
2 utente1:~$ ls
3 file1      file3
4 file2
```

► Copiare un file o una directory (cp)

La copia di un file si esegue con il comando `cp` (copy) che accetta due parametri: il nome del file da copiare (`file1` nell'esempio) e il nome che assumerà la copia (`copy_file1`). Il comando

```
1 utente1:~$ cp file1 copy_file1
```

assume che `file1` esista e crea un file `copy_file` il cui contenuto è identico a quello di `file1`. Aggiungendo l'opzione `-r` è possibile copiare una directory e il suo contenuto.