

## 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

### Esempio

Username: lcng07

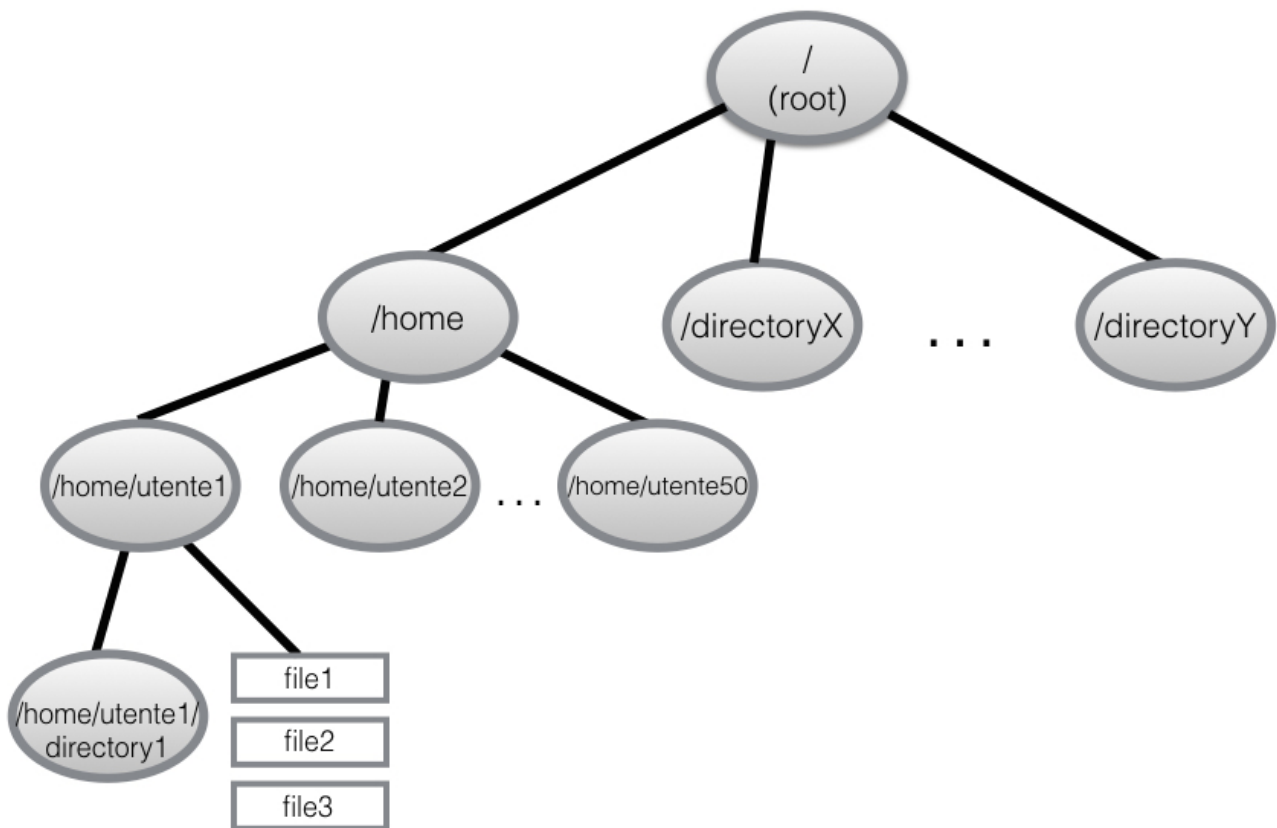
Password: lcng07

## 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 "/" (*slash*).

## *Esempio di struttura gerarchica delle cartelle*



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.

## 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 e' rappresentato dal carattere "\_" o da un rettangolino lampeggiante). Nel prompt possono essere indicate alcune informazioni come ad esempio il nome dell'utente

```
utente1:~$
```

## Visualizzazione della directory corrente (pwd)

Il comando `pwd` visualizza la posizione attuale dell'utente nella gerarchia nell'area del disco assegnata

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

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

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

```
utente1:~$ pwd
/home/utente1
utente1:~$ ls
directory1  file2
file1      file3
utente1:~$
```

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

- `ls -a` mostra anche i file nascosti
- `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

```
utente1:~$ ls -l a*
```

mostra, con tutti i dettagli, la lista dei file il cui nome inizia per `a`. In generale tutti i comandi Unix si compongono di una parola chiave (`ls`) cui seguono zero o più parametri (`a*`) 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`

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

## Creare una nuova directory (`mkdir`)

Per creare una directory, nella shell si usa il comando `mkdir`

```
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` seguito dal percorso di una directory permette di spostarsi all'interno di essa

```
utente1:~$ cd /home/utente1/directory1
utente1:~$ pwd
/home/utente1/directory1
```

usando il comando `cd ..` si sale di un livello nella gerarchia di cartelle

```
utente1:~$ pwd
/home/utente1/directory1
utente1:~$ cd ..
utente1:~$ pwd
/home/utente1
```

Per tornare nella directory principale (`/home/utente1`) si può dare il comando `cd ~`

Il simbolo `~` infatti rappresenta la propria home directory.

```
utente1:~$ pwd
/home/utente1/directory1
utente1:~$ mkdir my_dir
utente1:~$ cd my_dir
utente1:~$ pwd
/home/utente1/directory1/my_dir
utente1:~$ cd ~
utente1:~$ pwd
/home/utente1
```

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

```
utentel:~$ pwd
/home/utentel/directory1/my_dir
utentel:~$ cd ../../
utentel:~$ pwd
/home/utentel
```

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):

```
utentel:~$ pwd
/home/utentel
utentel:~$ ls
directory1  file2
file1       file3
utentel:~$ mv file1 nuovo_file
utentel:~$ ls
utentel:~$ directory1  file3
                     file2      nuovo_file
```

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

```
utentel:~$ ls
directory1  file2
file1       file3
utentel:~$ mkdir my_dir
utentel:~$ ls
utentel:~$ directory1  file1  file3
                     my_dir  file2
utentel:~$ mv my_dir directory1/
utentel:~$ cd directory1/
```

```
utentel:~$ ls  
my_dir
```

Nell'esempio precedente e' stata creata una cartella *my\_dir* e spostata nella *directory1*.

## Cancellare un file o una directory (rm)

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

```
utentel:~$ ls  
directory1  file2  
file1       file3  
utentel:~$ rm file1  
utentel:~$ ls  
utentel:~$ directory1  file3  
                        file2
```

Analogamente, con lo stesso comando e' 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`

```
utentel:~$rm -r directory1  
utentel:~$ ls  
file1       file3  
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

```
utentel:~$ cp file1 copy_file1
```

assume che file1 esista e crea un file copy\_file il cui contenuto è identico a quello di file 1. Aggiungendo l'opzione -r e' possibile copiare una directory e il suo contenuto.