Lezione 1

- *Esplorate il vostro file system. Qual è il pathname della vostra home directory?*
  Per esplorare il file system si possono usare i comandi seguenti:

  - `cd [pathname]` per spostarsi nella directory specificata da `[pathname]`;
  - `ls [options] [pathname]` per visualizzare i contenuti della directory specificata da `[pathname]` in accordo alle opzioni specificate da `[options]`.

  Per ottenere il pathname della propria home directory è sufficiente eseguire in sequenza i comandi seguenti:

```
> cd  # cd senza argomenti sposta l'utente
      # nella propria home directory
> pwd # stampa il pathname della directory
      # corrente (present working directory)
```

- *Visualizzate i file della vostra home directory ordinati in base alla data di ultima modifica.*

```
> cd
> ls -t # l'opzione t serve ad ordinare i file
        # in base alla data di ultima modifica
```

- *Che differenza c'è tra i comandi `cat, more, tail`?*

  - `cat` serve a concatenare file (o lo standard input) inviando il risultato dell'operazione allo standard output;
  - `more` è un filtro che serve a visualizzare su schermo un flusso di testo una pagina per volta;
  - `tail` stampa su standard output le ultime 10 righe dei file forniti come argomenti.

- *Un link simbolico può puntare ad un altro link che a sua volta punta ad un file?*
  Se è possibile, c'è un limite al numero di link simbolici che si possono avere in catena?
  Qual è lo svantaggio dei link simbolici rispetto ai link hard?
  Sì, un link simbolico può puntare ad un altro link che a sua volta punta ad un file:

```
> ln -s file_name link1
> ln -s link1 link2
```

  eseguendo il comando `cat link2` viene stampato a video il contenuto del file `file_name`.
  In generale, il limite al numero di link simbolici in catena in un sistema UNIX è 6:

```
> ln -s file_name link1
> ln -s link1 link2
> ln -s link2 link3
> ln -s link3 link4
> ln -s link4 link5
> ln -s link5 link6
> ln -s link6 link7
```

Eseguendo il comando `cat link7` si ottiene il messaggio d'errore `cat : link7: Too many symbolic links encountered`.
Tuttavia tale limite dipende dalla versione del sistema operativo: ad esempio, su alcune distribuzioni Linux (e.g., Ubuntu 12) il numero massimo di link simbolici in catena è 40, mentre in Mac OS X 10.6.8 è 32.
Lo svantaggio dei link simbolici rispetto ai link hard è la minore efficienza quando si vuole accedere al file puntato dal link. Infatti, per accedere al contenuto di un file bisogna conoscere il relativo numero di inode. Quest'ultimo è accessibile direttamente tramite un hard link (associazione esplicita fra nome del link e numero di inode), mentre bisogna compiere un passo ulteriore nel caso di un link simbolico, essendo quest'ultimo un file di testo (con un differente numero di inode) contenente il pathname del file puntato (si veda la figura nel lucido intitolato "Inode e link" della Lezione 1).

- *Trovate un modo per ottenere l'elenco delle subdirectory contenute ricorsivamente nella vostra home.*

```
> cd
> ls -R
```

- *Trovate due modi diversi per creare un file (suggerimento: consultare la man page del comando `touch`).*
  - utilizzare un editor;
  - utilizzare il comando `touch nome_file`; infatti `touch` provoca l'aggiornamento della data/ora di ultima modifica a quella corrente se il file (fornito come argomento) esiste già, mentre se quest'ultimo non esiste lo crea vuoto;
  - esiste un terzo modo di creare un nuovo file, utilizzando il comando `cat >nome_file` o `cat - > nome_file` (dalla man page di `cat`) e la redirezione dello standard output (carattere `>`).

- *I seguenti comandi che effetto producono? Perché?*

```
> cd
> mkdir d1
> chmod 444 d1
> cd d1
```

  - `cd` cambia la directory corrente nella directory home dell'utente;
  - `mkdir d1` crea la directory `d1` nella directory corrente (a meno che non esista già una directory con tale nome);

– `chmod 444 d1` imposta i permessi di `d1` a `r--r--r--`;
– `cd d1` fallisce provocando la visualizzazione del messaggio di errore `d1: Permission denied` in quanto per "entrare" in una directory è necessario possedere il permesso di esecuzione (`x`) su di essa.