

# Laboratorio Multimediale

## Lezione n. 2

Corso di Laurea in Matematica, a.a. 2006-2007

16 ottobre 2006

### Il filesystem

Il *filesystem* è una struttura dati che permette di organizzare grandi quantità di informazioni e di potervi accedere velocemente.

L'unità di base di un filesystem è il *file* (fascicolo, documento). Un *file* è una sequenza di *byte*. La più semplice interpretazione dell'informazione contenuta in un file si ottiene associando ad ogni byte un carattere (lettere, cifre, punteggiatura) tramite la tabella dei *codici ASCII*. In questo modo il file rappresenta un testo e viene quindi chiamato *file di testo*. In alternativa il file viene chiamato *file binario* e per accedere al suo contenuto bisogna utilizzare un programma che interpreti correttamente le informazioni contenute in esso.

Un filesystem può contenere moltissimi file in una struttura organizzata ad albero. Le *foglie* dell'albero sono i files, mentre i *nodi* vengono chiamati *directory* (archivio). In pratica ogni file è contenuto in una *directory*, e ogni *directory* può contenere file e anche altre *directory* (chiamate *sottodirectory*). Ad ogni file e ad ogni *directory* viene associato un nome, che dev'essere unico all'interno della *directory* che le contiene.

Nel sistema operativo GNU-Linux, la *directory* di base (la *radice* dell'albero) viene chiamata *root* e viene identificata dal nome `/` (una singola barra). Il nome *completo* di un file o di una *directory*, è la sequenza dei nomi delle *directory* che contengono il file, a partire dalla *root*, separati con delle barre `/`. Ad esempio se il file `tesi.tex` si trova nella *directory* `luca` che a sua volta si trova nella *directory* `home` del computer, il nome completo del file è `/home/luca/tesi.tex`. Il nome completo di un file o di una *directory* viene usualmente chiamato *path* (percorso).

Nei sistemi multiutente (come GNU-Linux) ad ogni utente viene assegnata una *directory* personale chiamata *home directory* o più brevemente *home*. In questa *directory* l'utente può memorizzare i propri file con la sicurezza che gli altri utenti non potranno nè cancellarli nè leggerli senza il suo permesso (si veda la sezione *permessi dei file*). Inoltre le macchine del nostro Dipartimento condividono la parte del filesystem che comprende le *home directory*. In questo modo è possibile accedere alla propria *home* da qualunque computer del Dipartimento.

Tramite l'utilizzo di una shell, si può accedere facilmente al filesystem. La shell ha il concetto di *directory corrente di lavoro*, utile per poter accedere velocemente ai files contenuti in tale *directory*, senza doverne specificare il nome completo. Quando viene specificato un nome di file che non inizia con la

barra /, si presume che il file sia contenuto nella directory corrente. Ad esempio, se la directory corrente è `/home/luca`, il file `tesi.tex` identifica il file `/home/luca/tesi.tex`. Se invece la directory corrente fosse `/home`, per riferirsi allo stesso file si potrebbe utilizzare il nome `luca/tesi.tex`. I nomi di files che non iniziano con una barra, vengono chiamati nomi (o path) relativi (in quanto identificano un file a partire dalla directory corrente). Quando si avvia la shell, la directory corrente coincide con la propria home directory. Se si cambia directory corrente si può velocemente tornare alla home dando il comando `cd`.

Il nome speciale `..` viene utilizzato per identificare la directory superiore. Per esempio il nome `/home/luca/../lettera.txt` identifica il file `/home/lettera.txt`. Analogamente, se la directory corrente è `/home/luca`, il nome `../lettera.txt` identifica il file `/home/lettera.txt`.

directory ..

## Comandi della shell

Ricordiamo che per accedere alla shell, in modalità grafica, bisogna premere sul pulsante che rappresenta uno schermo con una conchiglia (shell appunto).

`exit` chiude la shell.

`ls` elenca i files della directory corrente.

`ls -al` elenca i files (anche i files nascosti) dando maggiori dettagli sulle dimensioni e i permessi.

`ls -l` elenca i files su una unica colonna.

`pwd` visualizza il nome della directory di lavoro corrente (*print working directory*).

`cd dir` entra nella directory specificata (cambiando la directory corrente di lavoro). Se *dir* non è specificato torna alla propria home.

`cd ..` torna alla directory contenente la directory corrente.

`mkdir nome_dir` Crea una sottodirectory nella directory corrente.

`cp origine destinazione` fa una copia del file *origine* con il nome *destinazione*. Se *destinazione* è una directory, copia il file nella directory.

`mv origine destinazione` sposta il file. Si utilizza come il comando `cp`, ma il file di partenza viene cancellato. In particolare può essere utilizzato per cambiare nome ad un file.

`rm file` cancella il file specificato.

`rmdir directory` cancella la directory specificata (ma solo se è vuota).

`scp origine destinazione` copia i files come `cp` ma permette la copia da un computer ad un altro. I files possono essere infatti specificati nella forma `username@hostname:filename`.

`file nomefile` dà informazioni sul possibile tipo del file specificato (se è un file di testo, un file  $\text{T}_{\text{E}}\text{X}$ , un'immagine, un file compresso o altro...)

`less nomefile` visualizza il file sullo schermo, una pagina alla volta. Per terminare bisogna premere il tasto `q`. Per passare alle pagine precedente o successiva premere i tasti `PgUp` e `PgDown`. Per cercare una parola nel testo premere il tasto `/`, scrivere la parola da cercare e premere `Enter`.

`man comando` visualizza il manuale del comando specificato. Si utilizzano gli stessi tasti del comando `less` per visualizzare il testo. Si può anche utilizzare la forma `man -k parola` che elenca tutti i comandi che contengono la *parola* nella loro descrizione.

`echo frase` scrive su schermo la frase specificata.

`cat elenco files` scrive sullo schermo i files specificati, uno di seguito all'altro. Se nessun file viene specificato scrive in output l'input proveniente dalla tastiera.

`sort nomefile` scrive il file specificato elencando le righe in ordine alfabetico.

`tr from to` serve per modificare i caratteri in input. Si veda il manuale `man tr`.

`grep parola file` serve a selezionare le righe in input che contengono una determinata parola (si veda il manuale).

`cut -f colonne -d separatore` serve a selezionare alcune colonne dell'input (si veda manuale).

`paste file1 ... fileN` incolla i file per colonne (si veda manuale).

`tar`, `gzip`, `gunzip`, `zip`, `unzip` sono i comandi per comprimere e per impacchettare uno o più file.

## Come comporre tra loro i comandi della shell

I comandi della shell possono essere combinati tra loro per svolgere compiti più complessi. Molti comandi funzionano come dei *filtri*, prendono un *input*, lo modificano, e ritornano un *output*. Normalmente questi comandi prendono l'input dalla tastiera e mandano l'output al terminale. È però possibile *redirigere* l'input e l'output su file oppure concatenare l'output di un comando con l'input del successivo:

**redirezione output:** con la sintassi `comando > file` si manda l'output del comando sul file specificato.

**redirezione input:** con la sintassi `comando < file` si esegue il comando con l'input preso dal file specificato.

**pipe:** con la sintassi `comando | comando` l'output del primo comando viene preso come input dal secondo.

Esempi. Il comando `date > data.txt` scrive nel file `data.txt` la data odierna. Il comando `sort file1 > file2` ordina il file `file1` e mette il risultato in `file2`. Il comando `echo ciao | tr o u` scrive `ciau` sullo schermo.

## Come gestire più comandi contemporaneamente

È possibile mantenere in esecuzione più comandi contemporaneamente. Per lasciare un comando in esecuzione in *background* bisogna far seguire al comando il carattere `&`. In genere conviene eseguire in background ogni comando che apre una finestra (come `mozilla`, `emacs`...), in modo che la *shell* rimanga utilizzabile in contemporanea per eseguire altri comandi. Il comando `jobs` elenca i programmi in background. Se un programma è in esecuzione in *foreground* può essere interrotto premendo i tasti `Ctrl C`. Premendo i tasti `Ctrl Z`, il comando viene sospeso (*suspended*). Con il comando `bg` il comando sospeso viene riavviato in *background*, con `fg` il comando viene riavviato in *foreground*.

## Come gestire i permessi dei files

In un sistema operativo Unix, ad ogni file vengono associati dei *permessi* che indicano chi può leggere o scrivere il file. I file personali di un utente sono generalmente modificabili solo dall'utente stesso. Può essere però utile rendere accessibili in lettura i propri files. Usualmente i permessi si visualizzano con `ls -l` e si cambiano con `chmod`. Il sistema utilizzato al Dipartimento di Matematica, invece, è più complesso. La gestione dei permessi dei files avviene tramite il comando `fs`.

`fs help` dà un elenco di tutti i comandi `fs`.

`fs la directory` per visualizzare i permessi della directory specificata.

`fs setacl -dir directory -acl permessi` cambia i permessi della directory specificata. Ad esempio i permessi `system:anyuser rl` permettono la lettura dei files a tutti gli utenti.

## La posta elettronica

### Nozioni generali

Un messaggio di posta elettronica (*email* o *E-mail*), ricalca in gran parte una normale lettera di posta ordinaria. L'*email* si suddivide in *head* (intestazione) e *body* (corpo del messaggio). Il *body* è il messaggio vero e proprio, e consiste in un testo libero. Informazioni *sul* messaggio, vengono invece inserite nella intestazione. Tra queste si individuano in particolare le informazioni che nella posta ordinaria vengono scritte sulla busta.

Ogni informazione che compare nella intestazione è preceduta da un nome che ne identifica il significato. In particolare:

**From:** indica l'indirizzo del mittente. Come ogni indirizzo che compare nell'intestazione, può essere scritto in due forme diverse: `user@host` oppure `Nome <user@host>`.

**To:** indica il destinatario o i destinatari del messaggio. Più indirizzi possono essere specificati separandoli con una virgola.

**Subject:** indica l'oggetto del messaggio. Viene scritto dall'autore per riassumere brevemente il contenuto del messaggio.

**Date:** indica la data di spedizione. Questa informazione viene aggiunta automaticamente dal computer.

**Cc:** (*carbon copy*) indica uno o più indirizzi che ricevono il messaggio *per conoscenza*.

**Bcc:** (*blind carbon copy*) come il Cc solo che gli altri destinatari del messaggio non sapranno che il messaggio è stato spedito anche a questo indirizzo. Può essere utile quando si spedisce un messaggio a molte persone (*mailing list*) e non si vuole che ogni persona venga a conoscenza dell'indirizzo di tutte le altre.

**Message-Id:** viene aggiunto automaticamente per identificare univocamente il messaggio.

**Reply-To:** se presente specifica un indirizzo, diverso dal From, a cui si vuole che vengano spedite le risposte.

**Fcc:** indica un file (o mail folder) nel quale salvare una copia del messaggio inviato.

Gli indirizzi *email* sono stringhe della forma *username@host*. Lo *host* è il nome pubblico di un server (computer) collegato ad internet che gestisce il protocollo di posta elettronica SMTP (*simple mail transfer protocol*). Lo *username* è un identificativo univoco che permette all'*host* di identificare un utente. Nel caso del nostro laboratorio lo *username* di posta elettronica coincide con lo username utilizzato per il login. Il nome dello *host* che gestisce la posta elettronica è `students.math.unifi.it`. Ad esempio l'indirizzo *email* dell'utente `m06paoros` è `m06paoros@students.math.unifi.it`.

Oltre al testo del messaggio, che può essere solo un testo semplice (formato da codici ASCII standard), è possibile aggiungere al messaggio degli *attachment* (allegati). Ogni allegato è un intero file, che viene inserito nel messaggio con una speciale codifica. Chi riceve il messaggio può accedere ad i file allegati, visualizzandoli direttamente oppure salvandoli sul proprio computer.

Il software utilizzato per leggere e inviare la posta elettronica si chiama *mail client*. Sulle macchine del dipartimento si possono utilizzare diversi client, tra i quali ricordiamo: thunderbird (interfaccia grafica), mutt (interfaccia testuale), mail (riga di comando).

In alternativa si può accedere alla propria posta, da qualunque computer collegato ad internet, collegandosi all'indirizzo `http://webmail.math.unifi.it/`.

## Netiquette

Segnaliamo alcune considerazioni importanti per un corretto utilizzo della posta elettronica.

- Non è necessario inserire nel testo del messaggio le informazioni che sono già contenute nell'intestazione (come la data, il proprio indirizzo email, il destinatario).
- Il **Subject** va sempre specificato, e va scelto con ponderazione. Questo aiuterà chi riceve il messaggio ad organizzare meglio la propria posta elettronica.

- Trattare gli indirizzi *email* come dati sensibili (alla stregua di un numero di telefono).
- Non scrivere il messaggio utilizzando TUTTE MAIUSCOLE. Questo infatti verrebbe interpretato come un *urlare*.
- Quando si risponde ad un messaggio può essere utile riprendere (*quotare*) le parti del messaggio originale al quale si risponde. Bisogna però eliminare tutte le parti del messaggio originale che non sono di interesse, altrimenti dopo ogni passaggio il testo del messaggio raddoppierebbe di dimensioni.
- Non utilizzare l'HTML o formati diversi dal testo semplice per il corpo del messaggio. Anche l'utilizzo di estensioni ai caratteri ASCII (ad esempio le lettere accentate) possono risultare non leggibili ad alcuni destinatari. Se necessario inserire documenti in altri formati come allegato, ma non assumere mai che il destinatario possa visualizzare immediatamente tali formati.
- Si noti alcune persone sono infastidite dai messaggi pubblicitari o dalle cosiddette catene di S. Antonio. Dunque prima di spedire un messaggio a molte persone, accertarsi che il messaggio sia pertinente, e che sia di interesse a tutte le persone che si contattano. In particolare se si manda ad altri un messaggio ricevuto (*forward*), è necessario verificare la validità del messaggio ricevuto.
- Se si riceve un messaggio che era indirizzato a molte persone, fare attenzione quando si intende rispondere al messaggio. Spesso infatti capita di spedire il messaggio a tutti i destinatari quando invece sarebbe più corretto spedire il messaggio solo a chi lo ha inviato.
- Non fidarsi della provenienza (campo **From:**) dei messaggi email. Le intestazioni del messaggio potrebbero infatti essere state *forgiate* in maniera malevola. In particolare, se si riceve un messaggio indesiderato o fastidioso, accertarsi, prima di rispondere, che la provenienza sia autentica.
- Non fidarsi del buon funzionamento della posta elettronica. Può succedere che un messaggio inviato non venga, per vari motivi, letto dal destinatario. Controllare sempre eventuali messaggi di errore spediti dal *mail server*. Ricordarsi che tali messaggi di errore possono arrivare anche alcuni giorni dopo l'invio del messaggio. D'altro canto se ci arriva un messaggio importante può essere utile rispondere subito, per notificare al mittente che abbiamo ricevuto il messaggio.

## Compito odierno

1. Chi non l'avesse già fatto la lezione scorsa, deve cambiare la propria password.
2. Ogni studente deve segnalare la propria presenza dalle pagine del Laboratorio Multimediale.
3. Creare una directory chiamata `lezione2` nella propria *home directory*. Tutti i file che vi si chiede di creare nei punti seguenti, andranno messi in questa nuova directory. Comandi utili: `mkdir`, `cd`, `ls`.
4. Nella directory `lezione2` creare una struttura ad albero contenente le seguenti directory: `animali`, `vegetali`, `pesci`, `uccelli`, `mammiferi`, `piante` e i seguenti file: `ghepardo.html`, `elefante.html`, `triglia.html`, `pinguino.html`, `delfino.html`, `cipresso.html` in modo coerente con la classificazione delle specie. I file HTML devono contenere una pagina con la descrizione della corrispondente specie (da reperire su internet).
5. Determinare il nome completo della propria *home directory*, ed esplorare le directory precedenti nella gerarchia. Creare nella directory `lezione2` un file chiamato `compagni05.txt` contenente l'elenco degli *username* di tutti gli studenti del secondo anno (immatricolati nel 2005). *Facoltativo*: mettere gli *username* in ordine alfabetico secondo il cognome. Comandi utili: `pwd`, `ls -l`, `cd ..`, `sort`, `cut`, `paste`, *redirezione dell'output*.
6. *Facoltativo*. Dalle pagine del Laboratorio Multimediale, alla voce `materiale`, scaricare il file `comuni.txt` e salvarlo nella directory `lezione2`. Creare il file `lombardia.txt` contenente i nomi e il codice di avviamento postale di tutti i comuni della lombardia. Ordinare il file secondo il nome del comune. Comandi utili: `grep`, `cut`, `sort`, *redirezione dell'output*.
7. Spedire un messaggio di saluto a tutti i compagni di corso e in cc a `paolini@math.unifi.it`. Cosa succede se si scrive un messaggio ad un indirizzo errato? Fare una prova. Leggere i messaggi ricevuti dagli altri compagni.
8. Nella propria home directory (il comando `cd` senza argomenti vi porta nella home) dare il comando

```
tar czvf lezione2.tgz lezione2
```

questo comando crea il file compresso `lezione2.tgz` che contiene al suo interno tutti i file e le directory contenuti in `lezione2` (le opzioni `czvf` significano: `c`=create, `z`=zip, `v`=verbose, `f` *filename*=file di output). Spedire questo file come *attachment* a tutti i componenti del proprio gruppo (alle persone che lavorano allo stesso computer), in modo che anche gli altri abbiano una copia del lavoro fatto. Chi riceve il file lo può espandere dando il comando inverso `tar xzvf lezione2.tgz` (`x`=expand, `z`=zip, `v`=verbose, `f`=file). Inviare una copia dello stesso file al docente, utilizzando il servizio apposito che si trova in fondo alla pagina delle presenze.