

Laboratorio Multimediale

Lezione n. 2, sezione A

Corso di Laurea in Matematica, a.a. 2005-2006

20 ottobre 2005

Compito odierno

1. Chi non l'avesse già fatto la lezione scorsa, deve cambiare la propria password.
2. Creare una directory chiamata `lezione2` nella propria *home directory*. Tutti i files che vi si chiede di creare nei punti seguenti, andranno messi in questa nuova directory.
3. Determinare il nome completo della propria *home directory*, ed esplorare le directory precedenti nella gerarchia. Creare un file chiamato `compagni.txt` contenente l'elenco degli *username* di tutti i propri compagni d'anno. Mettere gli *username* in ordine alfabetico secondo il cognome.
4. Ricopiare il testo riportato nell'Appendice in un file chiamato `testo.txt`. Fare una copia del file appena creato, per non rischiare di perderlo nelle operazioni seguenti.
5. Creare il file `tistu.txt` (nella directory `lezione2`) in cui il testo del file `testo.txt` viene modificato come segue: ad ogni occorrenza della lettera `a` sostituire la lettera `e`, alla lettera `e` sostituire la lettera `i`, alla `i` la `o`, alla `o` la `u` e alla `u` la `a`.
6. Tramite la pagina web delle presenze, inviare i files: `compagni.txt`, `testo.txt` e `tistu.txt`.
7. Tutti i componenti del gruppo devono cercare di copiare la directory `lezione2` e tutti i files contenuti, nella propria *home directory*.

Comandi della shell

`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.

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

`cd dir` entra nella directory specificata (cambiando la directory corrente).

`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.

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

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

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

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

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 ogni comando che apre una finestra (come `mozilla`, `emacs`...) conviene eseguirlo in *background*, 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* (non in *background*), può essere interrotto premendo i tasti `Ctrl` `c`. Premendo i tasti `Ctrl` `z`, il comando viene sospeso (*suspended*). Con il comando `bg` il comando viene riavviato in *background*, con il comando `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 ottenere 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.

L'editor di testi emacs

Emacs è un editor di testi programmabile. Per avviarlo dare il comando **emacs** (scrivere **emacs &** se si è in modalità grafica). Si può anche dare il comando **emacs nomefile** per avviare **emacs** e aprire immediatamente il file specificato.

I comandi che possono essere utilizzati all'interno di **emacs** utilizzano generalmente una sequenza di caratteri speciale. Si indicherà con **C-x** la sequenza **Ctrl x**. Con **M-x** si intende invece la sequenza **Esc x** (il tasto **Esc** non va premuto prima, e non in contemporanea col tasto **x**).

Un elenco dettagliato dei comandi di **emacs** può essere trovato nel file **EmacsDoc**, reperibile nelle pagine del corso, alla voce **materiale**. Se si sbaglia a digitare un comando, premere **C-g** per annullare.

comandi di base **C-x C-c** termina emacs. Se il file aperto non è stato salvato chiede se si intende salvarlo.

C-x C-f richiede l'apertura di un nuovo file nella finestra corrente.

C-x C-s salva le modifiche apportate al file corrente.

C-x C-w chiede il nome del file in cui salvare il testo.

C-x C-i inserisce nel testo il contenuto di un file.

selezione e copia del testo Molti comandi agiscono su una regione di testo. La regione viene delimitata dalla posizione del *mark* e quella del cursore.

C-spazio mette il *mark* nella posizione corrente del cursore.

C-w taglia la regione tra il *mark* e il cursore.

M-w copia la regione.

C-y incolla il testo tagliato o copiato in precedenza.

C-k taglia il testo dalla posizione corrente fino alla fine della riga.

C-x r k taglia il rettangolo delimitato dal *mark* e dal cursore.

C-x r y incolla il rettangolo tagliato in precedenza.

altri comandi **C-s** cerca una parola nel testo.

M-x replace-string cerca tutte le occorrenze di una parola e le rimpiazza con un'altra parola specificata.

M-x query-replace-string come **replace-string** ma chiede conferma per ogni sostituzione.

M-x sort-lines mette in ordine alfabetico le righe del testo.

Appendice

29 maggio 1832. Evariste Galois non ha ancora ventun'anni, ma sente che non supererà l'indomani. Prende carta e penna e, tra le lettere, ne lascia una all'amico Auguste Chevalier a dir poco sconcertante. Nelle parole di esordio non ci sono le ragioni del duello che avrebbe affrontato di lì a poche ore, né i ricordi di una vita breve e intensa. Galois esordisce con ciò che per lui è più urgente in quel momento: la matematica.

"Mio caro amico, ho fatto in analisi molte cose nuove. Le une riguardano la teoria delle equazioni, le altre le funzioni integrali".

Nella sua breve esistenza appassionata e istintiva, raramente Evariste aveva dimostrato di possedere una tale dose di freddezza e distacco dalle emozioni. Infatti, se non fosse per il tono colloquiale e per il dramma imminente che col senno di poi leggiamo tra le righe, potremmo scambiare la lettera a Chevalier per un articolo di una rivista di matematica, o almeno di una sua minuta. In effetti in parte lo è: teoremi, dimostrazioni, passaggi, sviluppi di idee molto complesse, si susseguono senza lasciare trasparire i pensieri di un condannato a morte. Eccetto per quel cupo presagio vicino alla conclusione: Ma io non ho tempo... Poche ore dopo, Galois verrà ferito all'addome e morirà di peritonite acuta il 31 maggio 1832.

Arrestato in seguito a nuovi tumulti, Galois continuò i suoi studi dal carcere, dove rimase rinchiuso per alcuni mesi. Non sono chiari i motivi che lo spinsero a rinunciare alla vita e alla matematica, accettando quel duello: forse una donna o forse la politica. La lettera a Chevalier non ci aiuta, prima di concludere Galois insiste ancora sulla matematica: "Mi sono spesso azzardato nella mia vita ad avanzare proposizioni delle quali non ero sicuro; ma tutto quello che ho scritto qui è da quasi un anno nella mia testa ed è troppo nel mio interesse non sbagliarmi perché mi si sospetti di aver enunciato dei teoremi dei quali non avrei la dimostrazione completa".