

> vi

Vi troverete di fronte una schermata simile alla figura in alto che, in tutta sincerità, ai nostri occhi di utenti del terzo millennio, abituati ad icone ed effetti speciali, tutto sembra tranne che un editor degno di rispetto. Un breve messaggio di presentazione e degli strani simboli (“~”, il loro nome è tilde) che stanno ad indicare le varie righe, rappresentano lo spartano benvenuto di questo editor al “malcapitato” utente di turno.

Probabilmente il vostro primo impulso ora è quello di provare a scrivere qualcosa, giusto per vedere quello che accade... non fatelo, prima c'è qualcosa che dovete sapere. A differenza della stragrande maggioranza degli editor moderni, vi presenta due modalità di funzionamento distinte:

1. **Insert mode (modalità di inserimento)**, dove diventa possibile scrivere tutto quello che si desidera.
2. **Command mode (modalità di comando)**, utilizzata per dare all'editor vari tipi di comandi.

Immaginate queste due modalità come un interruttore, quando questo è spento, potete scrivere normalmente caratteri, numeri e simboli di ogni genere, quando invece è acceso, l'editor si trasforma in una “plancia di controllo” dalla quale diventa possibile fare ogni genere di diavoleria sui vostri files usando particolari combinazioni di tasti.

Iniziamo quindi a fare subito qualche esperimento.

Appena avviato, vi è di default in modalità comando, digitate quindi quanto segue:

```
:q      <invio>
```

l'editor si chiuderà, riportandovi al prompt della shell.

La maggior parte dei comandi cominciano con il carattere “:”, essi vengono scritti automaticamente nella cosiddetta “last line” alla pressione di tale carattere. La lettera **q** altro non è che l'abbreviazione di “quit” che serve a chiudere il programma se non si sono apportate modifiche a qualche file.

Se invece vogliamo comunicare al programma che si intende uscire senza salvare i cambiamenti apportati ad un file, bisogna forzare l'uscita digitando il comando:

```
:q!     <invio>
```

A.3 SWITCH:

Dalla modalità comando a quella inserimento e viceversa

Lanciate nuovamente vi e premete semplicemente il tasto “i”. Nella parte bassa dello schermo comparirà la scritta -INSERT- : siete passati dalla modalità comando a quella di inserimento. Provate adesso a scrivere qualche frase, a cancellarla, a modificarla e vi accorgete che, fortunatamente, l'insert mode di vi non riserva particolari sorprese. Notate che anche scrivendo la sequenza “:q!”, l'editor rimarrà aperto poiché siete in modalità inserimento. Ma come si fa a entrare nuovamente in modalità comando?

Basta premere il tasto ESC ed ecco tornati alla modalità comando.

A.4 SALVARE UN FILE

Rientrate nella modalità di inserimento e proviamo scrivete qualcosa a piacimento, premete il tasto ESC per passare alla modalità comando e salviamo il nostro primo file di testo. Ecco i comandi:

```
:w nomefile
```

Questo comando salva il file nella directory di lavoro corrente (write), assegnandogli il nome “nomefile”. L'editor resta aperto per consentire di apportare ulteriori modifiche.

```
:w /temp/nomefile
```

Questo comando salva invece il file nella directory /temp.

Ora potreste uscire dall'editor e controllare che **vi** abbia davvero salvato il file. In realtà non c'è alcun bisogno di uscire dal programma per farlo, basta digitare semplicemente:

```
:! ls -l
```

Il comando verrà eseguito ed il suo output presentato direttamente su schermo.

E' possibile avvalersi di questa funzionalità con qualsiasi comando GNU/Linux. La sintassi, come avrete senz'altro capito, è la seguente:

```
:! comando
```

In seguito basterà semplicemente premere INVIO per tornare alla modalità comando.

A.5 APRIRE UN FILE

Ovviamente è possibile aprire un qualsiasi file di testo presente nel disco rigido in un istante, digitando il comando:

```
:e nomefile
```

Il modo più comodo e senz'altro più rapido di aprire un file è quello di lanciare l'editor **vi** specificando direttamente il file da aprire sul prompt del terminale:

```
> vi /temp/nomefile
```

Se il file "nomefile" non esiste, **vi** lo comunicherà attraverso la *last line*.

Il modo più comodo e rapido di uscire da un file esistente salvando le modifiche apportate è:

```
:wq
```

che dice all'editor di salvare le modifiche (write) ed uscire (quit), tutto in un sol colpo.

A.6 TAGLIA, COPIA ed INCOLLA

Notate che anche quando siete in modalità comando è possibile muovere il cursore all'interno del testo. Provate a porre il cursore in una qualsiasi riga, quindi digitate **dd**. La riga è scomparsa poiché tale comando serve appunto a cancellare la linea di testo nella quale è presente il cursore.

Se vi capita di cancellare qualcosa per errore? Nessun problema, **vi** è dotato anche della famosa funzione "undo", premete semplicemente il tasto **u** per annullare l'ultima azione compiuta (in questo caso la cancellazione della riga), premete il tasto **U** per annullare invece tutte le modifiche al file.

Ma in **vi** esistono anche le famose funzioni di: taglia - copia - incolla, senza usare il mouse!

Prima di tutto posizionatevi con il cursore nella riga che intendete copiare, quindi digitate **yy**. Avete così effettuato l'operazione di "copia" della riga corrente.

Per "incollare" il testo selezionato, portate il cursore in un qualsiasi punto del file e digitate **p**.

E' possibile effettuare il "taglia", digitando **dd** anziché **yy**.

Per riassumere: il comando **dd** cancella (o taglia) la riga corrente, **yy** la copia e **p** incolla la riga "in memoria" in un qualsiasi punto del testo.

Ovviamente è anche possibile cancellare, tagliare o copiare più di una riga alla volta: basterà scrivere il numero di righe che si intende selezionare prima di immettere il comando di cancellazione o copia. In parole povere, per tagliare (o cancellare) 5 righe, usate il comando **5dd**, per copiarne 12 scrivete **12yy** e così via.

A.7 FUNZIONE DI RICERCA TESTUALE

Per cercare una qualsiasi parola in avanti è sufficiente digitare, all'interno della modalità comando:

```
/ parola_da_cercare
```

Il cursore si porterà automaticamente nella prima occorrenza trovata. Se volete continuare a cercare altre occorrenze di questa parola nel testo, basterà premere il tasto n (next).

Per cercare invece una qualsiasi parola all'indietro è sufficiente digitare, all'interno della modalità comando:

? parola_da_cercare

A **vi** è inoltre possibile associare delle particolari opzioni, ne citeremo qui soltanto due:

- c comando Esegue un particolare comando all'avvio.
- R Edita il file in sola lettura.

A.8 TABELLA RIASSUNTIVA DEI PRINCIPALI COMANDI

Acquisterete presto la dimestichezza che vi serve grazie all'allenamento, **vi** possiede numerosissimi comandi e citarli tutti non rientra nello scopo di questo manuale, in basso viene mostrata una tabella riassuntiva con i comandi base, già visti e nuovi:

| Comando | Descrizione |
|--------------------|--|
| (invio) | Comincia una nuova riga (solo in modalità inserimento) |
| ESC | Torna alla modalità comando |
| i | Inserire testo prima del cursore |
| TAB | Inserisce una tabulazione |
| dd | Cancellare l'intera riga corrente (o taglia) |
| D | Cancellare dal cursore alla fine della riga (o taglia) |
| ndd | Cancellare n righe (o taglia) |
| dw | Cancellare una parola (o taglia) |
| dg | Cancellare fino alla fine del testo (o taglia) |
| G | Andare alla fine del file |
| 1G | Andare all'inizio del file (alla prima riga) |
| nG | Andare alla riga numero n |
| CTRL G | Visualizza il numero di riga corrente |
| :n | Si sposta alla riga numero n |
| R | Scrive in modalità sovrascrittura |
| u | Annulla l'ultima modifica (undo) |
| U | Annulla tutte le modifiche al file |
| x | Cancella il carattere sul cursore |
| ZZ | Esce da vi, salvando il file solamente se sono state apportate modifiche |
| :x | Come ZZ |
| :w | Salva il file |
| :wq | Salva ed esce dal programma |
| :w nomefile | Salva una copia |
| :w! | Salva il file ignorando le protezioni |
| :w! file | Sovrascrive il file |
| :q | Esce dal programma |
| :q! | Esce dal programma scartando le modifiche |
| yy | Copia l'intera riga corrente |
| nyy | Copia n righe |
| ye | Copia una parola |

| Comando | Descrizione |
|--------------------|---|
| p | Incolla l'ultimo testo copiato o tagliato dopo il cursore |
| / | Ricerca un pattern in avanti |
| ? | Ricerca un pattern all'indietro |
| : | L'editor attende un comando ex |
| ! | L'editor attende un comando shell |
| cc | Modifica una riga |
| cw | Modifica una parola |
| :r file | Inserisce il contenuto del file "file" dopo il cursore |
| :r! comando | Inserisce l'output del comando dopo la riga corrente |
| :! comando | Esegue il comando shell e restituisce il controllo |
| :so file | Legge ed esegue comandi letti da file "file" |

L'editor **vi** legge un file chiamato `~/.exrc`, all'inizio di ogni sessione di editing per settare certe impostazioni. I parametri per la modifica di questo file sono davvero troppi e non tratteremo questo argomento.

A.9 EDITOR E WORD PROCESSOR

Uno degli errori più comuni è quello di confondere gli editor di testo con i word processor. Queste due classi di programmi, apparentemente simili poichè servono entrambi per scrivere, sono in realtà diverse.

Con word processor si intende quella categoria di programmi che servono per scrivere testo formattato, cioè con varie opzioni di abbellimento come l'utilizzo del grassetto, dei colori, delle colonne multiple e che salvano i loro file in formati particolari che consentono di tener conto di tutte le modifiche di stile. Gli editor di testo, invece, non si occupano della formattazione e concentrano tutte le loro potenzialità nel trattamento puro delle stringhe di testo.

Per maggiori informazioni sull'editor e le sue opzioni, consultare il manuale online:

> **man vi**