

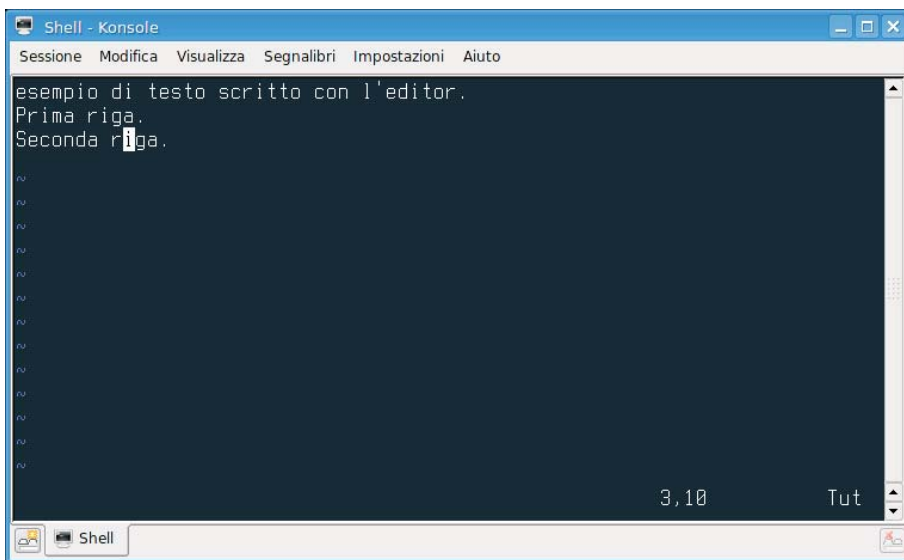
L'editor vi da linea comandi

L'editor con interfaccia a caratteri, più usato comunemente fin dalle prime versioni del sistema operativo Unix, è **vi**.

Il nome *vi* sta per *visual editor*, infatti è possibile vedere a video tutta la parte del file vicina alla linea corrente, nell'ambito della capacità del video medesimo (*full screen editor*). Nelle versioni recenti di Linux il programma si chiama **vim**.

Le caratteristiche principali di *vi* sono:

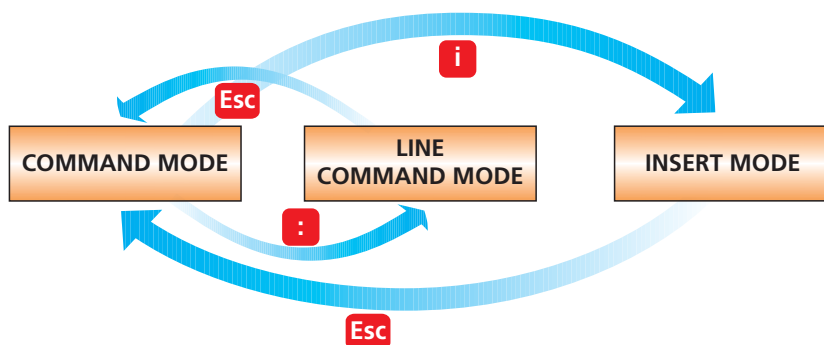
- è formato da un grande numero di comandi;
- i comandi sono brevi, quasi sempre costituiti da un solo carattere, e invisibili (cioè con assenza di eco sul video del comando battuto sulla tastiera);
- non possiede nessun comando di formattazione o stampa.



L'editor *vi* ha tre modalità di funzionamento:

- *command mode* è lo stato iniziale nel quale possono essere dati i comandi in modo invisibile;
- *line command mode* è lo stato in cui i comandi vengono scritti su una linea, l'ultima del video;
- *insert mode* è lo stato nel quale vengono inseriti nuovi caratteri o nuove parole di testo.

All'inizio il programma si trova in *command mode*; per passare all'*insert mode* si deve dare il comando *i*; per tornare al *command mode* si preme il tasto ESC; dal *command mode* si passa al *line command mode* con il tasto *:* (due punti).



L'editor *vi* viene attivato con:

```
vi prova
```

indicando il nome che si vuole assegnare al nuovo file; se il file *prova* esiste già viene visualizzato il suo contenuto e il file può essere modificato.

Altri formati di attivazione dell'editor *vi* sono:

```
vi nomefile/s
```

attiva *vi* accodando gli eventuali file indicati;

```
vi +n nomefile
```

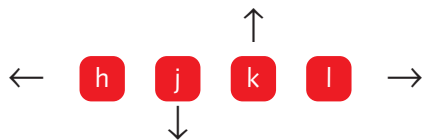
attiva *vi* con il cursore posizionato sulla linea indicata con il numero *n*;

```
vi +/parola nomefile
```

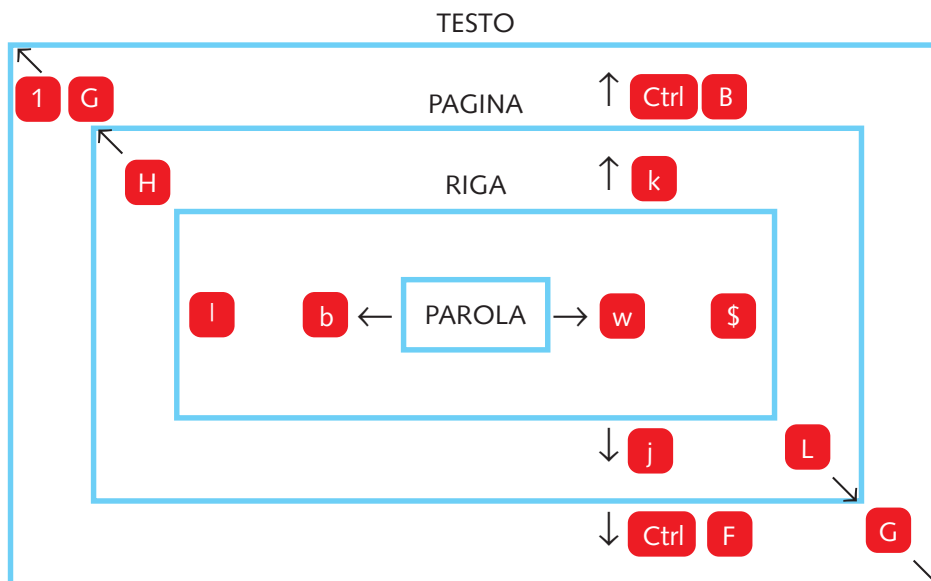
attiva *vi* con il cursore posizionato sulla linea contenente la *parola* indicata (alla prima occorrenza).

I comandi principali che possono essere dati in *command mode* sono illustrati nello schema seguente (per passare al modo comandi si deve premere il tasto ESC); si faccia molta attenzione alle differenze tra i caratteri in minuscolo e in maiuscolo, che assumono significati diversi quando vengono usati come comandi dell'editor in *command mode*.

Movimento del cursore



Movimento del cursore (parola, riga, pagina, testo)



Inserimento

- Carattere **I** a inizio riga **i** a sinistra **a** a destra **A** a fine riga
- Riga bianca **o** sotto la linea corrente **O** sopra la linea corrente

Cancellazione

- carattere sotto il cursore **x**
- parola **d w** a destra **d b** a sinistra
- riga corrente **d d**
- dal carattere a fine riga **D**
- dal carattere a fine schermo **d L**
- dal carattere a fine testo **d G**

Correzioni

- carattere **r**
- dal carattere in poi **R** fino alla pressione del tasto ESC
- parola **c w**
- riga **c c**

Blocchi

- Per copiare una riga
premere **Y** sulla riga corrente
muovere il cursore nel punto dove copiare la riga (senza premere ESC)
premere **p**
- Per copiare un blocco di righe:
premere **n Y**
(dove *n* indica il numero di righe da copiare a partire dalla riga corrente)
muovere il cursore nel punto dove copiare le righe (senza premere ESC)
premere **p**
- Per spostare una riga
premere **d d** sulla riga corrente
muovere il cursore nel punto dove spostare la riga (senza premere ESC)
premere **p**

- Per spostare un blocco di righe

premere **n** **d** **d**

(dove *n* indica il numero di righe da spostare a partire dalla riga corrente)

muovere il cursore nel punto dove spostare le righe (senza premere ESC)

premere **p**

- Per salvare su disco un blocco di righe

```
:x,y w nomefile
```

(in *line command mode*)

x,y indicano i numeri della prima e dell'ultima riga del blocco da scrivere sul disco; *w* indica l'operazione di scrittura (*write*), *nome* indica il nome da assegnare al file nel quale il blocco viene salvato.

- Per inserire un blocco di righe leggendolo da un file su disco:
muovere il cursore nel punto del testo dove si vuole inserire il blocco di righe; in *line command mode* scrivere

```
:r nomefile
```

dove *r* indica l'operazione di lettura (*read*) e *nomefile* è il file contenente le righe da inserire nel testo.

Ricerca e sostituzione

- ricerca **/** in avanti **?** all'indietro
- ripete la ricerca **n** nella stessa direzione **N** all'opposto
- sostituzione **:n1,n2 s/vecchia/nuova** (in *line command mode*)

dove *n1*, *n2* indicano il numero della prima riga e il numero dell'ultima riga per la sostituzione su una parte del testo; per la sostituzione di tutto il testo occorre mettere 1 al posto di *n1* e \$ al posto di *n2*.

Vecchia indica la parola o la parte di testo esistente che deve essere sostituita con *nuova*.

Per sostituire tutte le occorrenze all'interno della riga, occorre aggiungere */g* alla fine del comando:

```
:n1,n2 s/vecchia/nuova/g
```

Registrazione e uscita

- abbandona **:** **q** **!** (uscita senza registrare le modifiche)
- registra **:** **w** (salvataggio temporaneo)
- registra ed esce **:** **w** **q** oppure **Z** **Z**

Numerazione delle linee

(in *line command mode*)

```
:se nu
```

(abbreviazione di *set number*)

```
:se nonu
```

toglie la numerazione

Elimina l'effetto dell'ultimo comando

u (*undo*)

In *line command mode* (con il tasto :) si possono eseguire comandi Unix dall'interno dell'editor *vi* facendo precedere il comando dal carattere ! (punto esclamativo). Per esempio:

```
:!ls -l
```

dall'ambiente di *vi* è possibile ottenere la lista dei file presenti nella directory corrente.

Tutti i comandi possono essere preceduti dalla cardinalità del comando, cioè da un numero che indica quante volte il comando deve essere ripetuto.

Per esempio:

10dd

cancella 10 linee a partire dalla riga corrente.

L'efficacia del comando inoltre è relativa al punto dove si trova il cursore.