

EMACS (editor full screen)

Movimento cursore

freccia su (o CTRL-P)	riga precedente
freccia giù (o CTRL-N)	riga successiva
freccia destra (o CTRL-F)	carattere successivo
freccia sinistra (o CTRL-B)	carattere precedente
PgUp (o ESC-V)	pagina precedente
PgDown (o CTRL-V)	pagina successiva
CTRL-L	scrolla il testo ponendo la linea col cursore a centro schermo
ESC-F	parola successiva
ESC-B	parola precedente
CTRL-A	inizio riga
CTRL-E	fine riga
ESC-A	inizio frase
ESC-E	fine frase
ESC- <	inizio file
ESC- >	fine file

Editing

BS	cancella carattere precedente
CTRL-D	cancella carattere successivo
ESC- BS	cancella parola precedente
ESC-D	cancella parola successiva
CTRL-K	cancella dalla posizione del cursore a fine riga
ESC-K	cancella dalla posizione del cursore a fine frase
CTRL-Y	annulla l'ultima cancellazione
ESC-Y	annulla le ultime cancellazioni, in ordine

Varie

CTRL-U <i>n comando</i>	ripete <i>comando</i> per <i>n</i> volte
CTRL-G	elimina un prefisso di controllo, esce dall'help o da un loop, etc.
CTRL- - (o CTRL-X U)	annulla l'ultima azione
CTRL-Z	sospende Emacs e torna alla shell (%emacs per riprendere)
ESC-X "help"	apre l'help (se CTRL-H = BS serve per l'editing)
ESC-X "recover file"	aggiorna un file recuperando dall'ultimo salvataggio automatico

EMACS (continua)

Gestione File

CTRL-X CTRL-F	apre un file (o gli dà nome, o seleziona quel buffer se già aperto)
CTRL-X CTRL-S	salva i file aperti (che sono stati modificati, previa conferma)
CTRL-X CTRL-B	elenca i buffer attivi (file aperti, help, etc.)
CTRL-X CTRL-C	esce da Emacs (previo salvataggio dei file non salvati)

Gestione Finestre

CTRL-X 2	splitta in due la finestra attiva
CTRL-X 1	chiude tutte le finestre tranne quella attiva
CTRL-X O	cambia la finestra attiva (tipo <i>alt-tab</i>)
CTRL-X 4 CTRL-F	apre un file in una nuova finestra
ESC- CTRL-V	scrolla in giù il file nell'altra finestra

Ricerca e sostituzione

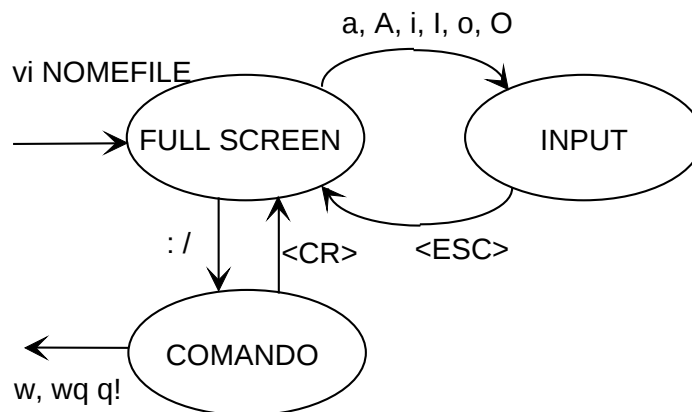
CTRL-S	ricerca in avanti (altri CTRL-S cercano il succ., INVIO termina)
CTRL-R	ricerca indietro (altri CTRL-R cercano il prec., INVIO termina)
ESC-X	“replace string”

EDITOR STANDARD DI UNIX: VI (VIsual display editor)

3 stati di VI:

- 1) lo stato comandi display (*full screen*);
- 2) lo stato riga comandi (*comando*);
- 3) lo stato di input (*input*).

Diagramma a stati del VI:



legenda: Maiuscole prima del cursore, minuscole dopo il cursore

comando per invocarlo **vi <nomefile>**

il file di nome <nomefile> viene aperto se esiste, creato altrimenti. Inizio editing ed accettazione comandi

CASO FILE ESISTENTE ==> vi prova.c

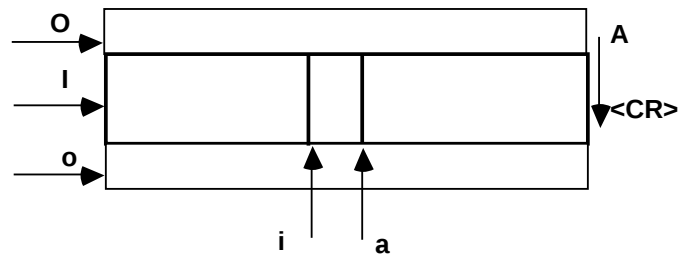
```
#include <stdio.h>

main(argc, argv)
int argc;
char ** argv;
{int i;
  for (i = 0; i < argc; i++)
    printf("argv[%d] = %s\n", argv[i]);
}
~
"prova.c"  9 lines, 138 characters
```

<====
RIGA
COMANDI

COMANDI di INPUT

Si passa da stato comandi a input in posizioni diverse



COMANDI FULL SCREEN

MOVIMENTO

di un carattere

quattro tasti sicuri (h,j,k,l)

← h ↓ j ↑ k → l

<blank> carattere successivo

<backspace> carattere precedente

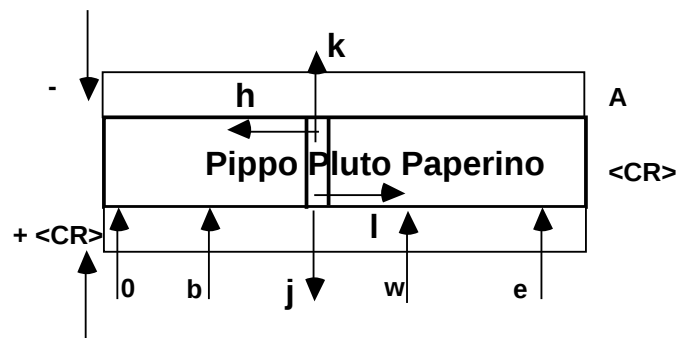
w una parola dopo

b una parola prima

e fine della parola corrente

<CR> di una riga

\$ fine della riga corrente



Alcuni comandi possono essere preceduti da un numero che indica il numero di volte che il comando deve essere ripetuto

Ad esempio: **3w** sposta il cursore in avanti di 3 parole

VISUALIZZAZIONE

visualizza videata seguente

<CTRL> F (indicato ^F)

visualizza videata precedente

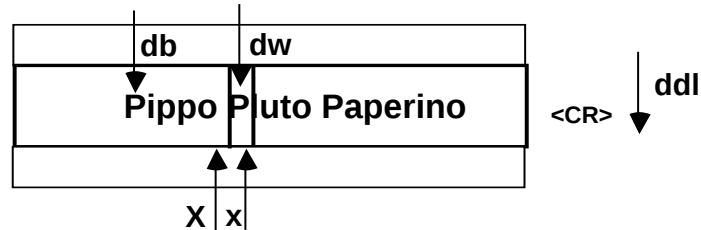
<CTRL> B (indicato ^B)

rinfresca il buffer (corrente)

<CTRL> L (indicato ^L)

DISTRUZIONE

di un carattere **x** (corrente), **X** (prima)
di una parola **dw** (dopo),
 db (prima)
di una riga **dd**
di tutto il resto della riga corrente **dS** o **D**



SOSTITUZIONE

di un carattere **r** <ch>
di un testo al posto del carattere **R** testo <ESC>
di un testo al posto della parola corrente **cw**testo <ESC>

AZIONI VARIE

ripete l'ultima azione **.**
annulla l'ultima azione eseguita **(undo)u**
annulla tutte le azioni sulla linea **U**
unisce la linea corrente con la seguente **J**
ripete la ricerca in avanti **n**
ripete la ricerca all'indietro **N**

COMANDI con uso di BUFFERIZZAZIONE

Esiste un buffer corrente senza nome, in cui sono anche memorizzate le ultime parti distrutte
copia la riga corrente in un buffer **nY nyy**
copia il buffer sopra la riga corrente **P**
copia il buffer dopo la riga corrente **p**
(maiuscolo prima, minuscolo dopo)
altri buffer (nomi da **a** a **z**, p.e. **f**) **"fnyy**
salvataggio e ripristino **"fnp "fnP**

Esempi:

a) COPIA DI UN INSIEME DI RIGHE

- posizionare il cursore sulla prima riga da copiare
- **<n>Y** copia <n> righe nel buffer
- spostare il cursore sulla riga sotto la quale copiare
- **p** inserisce le righe prelevate dal buffer

b) MOVIMENTO DI UN INSIEME DI RIGHE

- posizionare il cursore sulla prima riga da spostare
- **<n>dd** muove <n> righe nel buffer
- spostare il cursore sulla riga sotto la quale copiare
- **p** inserisce le righe prelevate dal buffer

COMANDI ULTERIORI

COMANDI della RIGA COMANDO (stato comando):

RICERCA

di una stringa specificata (avanti) /stringa
di una stringa (indietro) ?stringa

MOVIMENTO del cursore

su una linea specificata :numerolinea
ultima linea del file :\$

DISTRUZIONE

di linee del file :n,md linee da n a m

INTERAZIONE con il FILE SYSTEM

lettura ed inserimento di file alla posizione corrente	:r nomefile
lettura ed inserimento di file dopo la linea n	:nr nomefile
uscita da vi	:q
senza salvare	:q!
scrittura su file	:w nomefile
parti su file	:n,mw nomefile
aggiunte su file	:n,mw >> nomefile
scrittura su file superando i normali controlli (se possibile)	:w! nomefile
specifica di opzioni	:set opzione
visualizza i numeri di riga	:set number
elimina i numeri di riga	:set nonumber

ESECUZIONE di un qualunque comando di SHELL

escape ad uno shell

!:comando di shell>

Esempi di comandi:

a) SALVA ED ESCE DALL'EDITOR

- <ESC> esce dallo stato input
- :wq salva ed esce dall'editor

ATTENZIONE:

Se il file non consente la scrittura da parte dell'utente che sta usando il vi, l'editor riporterà un errore quando si tenta di scrivere sul file

b) ESCE DALL'EDITOR SENZA SALVARE

- <ESC> esce dallo stato input
- :q! esce dall'editor senza salvare

ATTENZIONE:

Se si usa :q senza aver salvato il file, l'editor avvisa

No write since last change (:quit! overrides)

Esistono moltissimi altri comandi: nonostante la scarsa leggibilità, **vi** è l'unico editor presente in tutti i sistemi UNIX.

Si consiglia di approfondirne la conoscenza.