

Vim

Luigi Carusillo
lgcrsl@email.it

I edizione: 25 settembre 2005
II edizione: 9 febbraio 2013

Indice

Introduzione	1
1 Configurazione	2
2 Aspetti generali	3
2.1 Componenti	3
2.2 Modalità operative	3
2.3 Schema di sessione	3
2.4 Creare un file	3
2.5 Aprire un file	3
2.6 Chiudere un file senza uscire da <i>Vim</i> (ripulire il buffer)	4
2.7 Chiudere un file e uscire da <i>Vim</i>	4
2.8 Aprire e chiudere due file	4
3 I comandi di <i>Vi</i>	4
3.1 Spostamento del cursore	5
3.2 Selezione	5
3.3 Copia	5
3.4 Modifica	6
3.5 Ricerca	6
3.6 Vari	6
3.7 Uscita	7
4 I comandi di <i>Ex</i>	7
4.1 Apertura di un altro file	7
4.2 Impostazione delle opzioni	7
4.3 Vari	7
4.4 Salvataggio	7
4.5 Uscita	7
Riferimenti bibliografici	7
Indirizzi	8

Introduzione

Vi (pronuncia: Vi-ai), sviluppato da William N. Joy, era un programma di videoscrittura su terminale, a tutto schermo e ad interfaccia testuale (cioè, i tasti avevano la doppia funzione di eseguire i comandi e di inserire il testo). *Vi* era basato anche su **Ex**, un programma di videoscrittura a riga singola.

Vim, di Bram Moolenaar, è la versione ampliata di *Vi*.

Origine dei termini

- Vi proviene da *visual interface editor* = programma di scrittura ad interfaccia visuale, cioè su schermo.
- Vim proviene da *visual improved* = vi migliorato.

Nomenclatura

Carattere: carattere pieno, della larghezza di una colonna, rappresentato da una lettera o un simbolo.

Spazio: carattere vuoto, della larghezza di una colonna, rappresentato da uno spazio.

Stringa: successione generica di caratteri, con o senza spazi al suo interno.

Parola: carattere singolo o successione di caratteri senza spazi al suo interno, chiusi da uno spazio.

Frase: parola singola o successione di parole, chiusa da un punto (., !, ?) e da uno spazio o una riga vuota.

Paragrafo: parola singola o successione di frasi chiusa da una riga vuota.

Spazio di tabulazione: successione di spazi singoli.

1 Configurazione

Il funzionamento di *Vim* è regolato da una serie di opzioni. Tutte le opzioni disponibili sono elencate e descritte nel file `/usr/share/vim/vim73/doc/options.txt`. Qui ne sono elencate e descritte alcune:

bk, backup	Crea un file di backup, cioè una copia di sicurezza del file, prima di sovrascrivere il file.
ic, ignorecase	Ignora la differenza tra maiuscole e minuscole durante la ricerca nel testo.
nu, number	Mostra a sinistra i numeri di riga.
ru, ruler	Indica la posizione del cursore sotto ogni finestra.
sm, showmatch	Quando viene inserita una parentesi chiusa, il cursore si sposta brevemente per mostrare la corrispondente parentesi aperta.
smd, showmode	Mostra nella riga di stato il tipo di inserimento, sia nella <i>modalità Vi</i> con i comandi di selezione e sostituzione che nella <i>modalità input</i> .
swf, swapfile	Crea un file di swap (= di scambio), cioè un file di backup provvisorio, per il buffer attuale.
tw=, textwidth=	Definisce la larghezza (cioè, il numero massimo di colonne) del testo: quando una riga supera la larghezza del testo, l'ultima parola va a capo.
wn, warn	Mostra un messaggio nella riga di stato, se si cerca di uscire da <i>Vim</i> senza salvare le modifiche.
ws, wrapscan	Quando la ricerca nel testo è giunta alla fine del file, ricomincia dall'inizio del file.
wa, writeany	Consente di salvare qualsiasi file.

Creare nella propria home directory il file `.vimrc` e inserirvi i comandi con le opzioni personalizzate. I comandi sono gli stessi descritti nella **Sottosez. 4.2** a pag. 7, ma non sono preceduti da « : » (il comando di accesso alla *modalità Ex*):

```
se opzione
se noopzione
se opzione=valore
```

Per esempio:

```
se nobk
se ic
se noswf
se tw=80
se wa
```

È possibile creare più file `.vimrc`, rispondenti ad esigenze diverse:

- il consueto file generico nella propria home directory;
- file specifici in altrettante directory ospitanti differenti tipi di testi.

2 Aspetti generali

2.1 Componenti

Vim appare sullo schermo suddiviso in due componenti:

- il buffer (che occupa la maggior parte dello schermo), in cui si inserisce il testo;
- la riga di stato (in fondo allo schermo), che mostra i comandi inseriti dall'utente e i messaggi di stato o di errore del programma.

2.2 Modalità operative

Vim opera in tre modalità:

- *modalità di comando di Vi* o *modalità Vi*: per i comandi, **Sez. 3** a pag. 4;
- *modalità di comando di Ex* o *modalità Ex*: per i comandi, **Sez. 4** a pag. 7;
- *modalità di inserimento del testo* o *modalità input*. L'unico comando proprio di questa modalità è **Esc**, che termina la *modalità input* e attiva la *modalità Vi*.

2.3 Schema di sessione

Una sessione di videoscrittura con *Vim* comprende tutte e tre le modalità operative.

1. *Vim* si avvia in modo predefinito nella *modalità Vi*.
2. La stesura del testo si ottiene dall'alternanza tra la *modalità Vi* e la *modalità input*.
 - a) Per inserire il testo nel buffer, si passa alla *modalità input* (principalmente con i comandi di inserimento, ma anche con comandi di sostituzione e cancellazione: **Sottosez. 3.4** a pag. 6).
 - b) Per modificare il testo, si torna alla *modalità Vi* (con **Esc**).
3. Per salvare e/o uscire, dalla *modalità Vi* si passa alla *modalità Ex* (con il comando « : » seguito da un comando specifico). Tuttavia, salvare e uscire contemporaneamente è possibile anche nella *modalità Vi*.

2.4 Creare un file

- Avviare *Vim* con il buffer predefinito (contenente alcune informazioni) e nominare, subito o in un secondo momento, il nuovo file:

```
$ vim
:w nuovo-file
```

Quando si decide di nominare il file, bisogna anche salvarlo.

- Avviare *Vim* con il buffer vuoto, relativo al nuovo file:

```
$ vim nuovo-file
```

Il file viene salvato subito.

2.5 Aprire un file

Secondo i casi:

```
$ vim file
```

oppure

```
$ cd /percorso
$ vim file
```

oppure

```
$ vim
:e /percorso/file
```

2.6 Chiudere un file senza uscire da *Vim* (ripulire il buffer)

- Se il file è rimasto immutato o è stato modificato e salvato prima:

```
:bd
```

- Se le modifiche non devono essere salvate:

```
:bd!
```

- Se le modifiche devono essere salvate:

```
:w  
:bd
```

2.7 Chiudere un file e uscire da *Vim*

- Se il file è rimasto immutato o è stato modificato e salvato prima:

```
:q
```

- Se le modifiche non devono essere salvate:

```
:q!
```

- Se le modifiche devono essere salvate:

```
ZZ oppure :wq oppure :x
```

2.8 Aprire e chiudere due file

- Per aprire due file:

```
$ vim primo-file  
:e secondo-file
```

- Per passare da un file all'altro (cioè, cambiare buffer):
 - se il file è rimasto immutato o è stato modificato e salvato prima:

```
:e#
```

- se le modifiche non devono essere salvate:

```
:e!#
```

- se le modifiche devono essere salvate:

```
:w  
:e#
```

- Per copiare da un file all'altro:

```
:e secondo-file oppure :e#  
V  
y  
:e#  
P
```

- Per chiudere i file e/o *Vim*, vedere la **Sottosez. 2.6** o la **Sottosez. 2.7** (secondo i casi).

3 I comandi di *Vi*

« *n* » è un numero che può precedere un comando per ripeterne l'azione.

3.1 Spostamento del cursore

Spostamento aspecifico, cioè senza riferimento a parti di file

↑	Nella stessa colonna della riga precedente.
↓	Nella stessa colonna della riga successiva.
←, Ritorno	Nella colonna di sinistra.
→, Spazio	Nella colonna di destra.
Pag↑	Alla schermata precedente.
Pag↓	Alla schermata successiva.

Spostamento specifico, cioè con riferimento a parti di file

b	<i>n</i>	All'inizio della parola corrente o della parola precedente.
e	<i>n</i>	Alla fine della parola corrente o della parola successiva.
w	<i>n</i>	All'inizio della parola successiva.
B	<i>n</i>	All'inizio della parola corrente o della parola precedente.
E	<i>n</i>	Alla fine della parola corrente o della parola successiva.
W	<i>n</i>	All'inizio della parola successiva.

-	<i>n</i>	All'inizio della riga precedente.
0 (zero)	<i>n</i>	All'inizio della riga corrente.
\$	<i>n</i>	Alla fine della riga corrente.
+, Invio	<i>n</i>	All'inizio della riga successiva.

(<i>n</i>	All'inizio della frase corrente o della frase precedente o alla fine del paragrafo precedente.
)	<i>n</i>	All'inizio della frase successiva o alla fine del paragrafo corrente.
{ (AltGr-7)	<i>n</i>	All'inizio del paragrafo corrente.
} (AltGr-0)	<i>n</i>	Alla fine del paragrafo corrente.

H	All'inizio della schermata.
nH	Alla riga <i>n</i> dall'inizio della schermata.
M	Al centro della schermata.
nL	Alla riga <i>n</i> dalla fine della schermata.
L	Alla fine della schermata.

1G	Alla prima riga del file.
nG	Alla riga <i>n</i> del file.
G	All'ultima riga del file.

I comandi **b**, **e**, **w** considerano come parole anche i segni di punteggiatura. I comandi **B**, **E**, **W** ignorano i segni di punteggiatura.

I comandi di spostamento specifico possono seguire i comandi di copia (**y**) e di cancellazione (**c**, **d**).

3.2 Selezione

v	Consente di selezionare il testo con i comandi di spostamento.
V	Seleziona un'intera riga e le righe precedenti o le righe successive con i comandi di spostamento relativi alle righe.

I comandi di selezione ripetuti annullano la selezione.

3.3 Copia

y	<i>n</i>	Copia il testo indicato dal comando di spostamento specifico immediatamente seguente (per esempio: yb , yB , y- , y(, y{ , yH , y1G).
		Copia il testo selezionato.
Y	<i>n</i>	Copia il testo dal cursore alla fine della riga.
yy	<i>n</i>	Copia un'intera riga.

Il testo copiato viene memorizzato per poter essere aggiunto (incollato) altrove.

3.4 Modifica

i	<i>n</i>	Inserisce il testo prima del cursore e attiva la <i>modalità input</i> .
a	<i>n</i>	Inserisce il testo dopo il cursore e attiva la <i>modalità input</i> .
l	<i>n</i>	Inserisce il testo all'inizio della riga e attiva la <i>modalità input</i> .
A	<i>n</i>	Inserisce il testo alla fine della riga e attiva la <i>modalità input</i> .
O	<i>n</i>	Inserisce il testo in una nuova riga vuota sopra la riga corrente e attiva la <i>modalità input</i> .
o	<i>n</i>	Inserisce il testo in una nuova riga vuota sotto la riga corrente e attiva la <i>modalità input</i> .

r	<i>n</i>	Sostituisce il carattere in corrispondenza del cursore.
R	<i>n</i>	Sostituisce una stringa nella <i>modalità input</i> .

s	<i>n</i>	Cancella un carattere e attiva la <i>modalità input</i> .
c	<i>n</i>	Cancella il testo indicato dal comando di spostamento specifico immediatamente seguente (per esempio: cb , cB , c- , c(, c{ , cH , cIG) e attiva la <i>modalità input</i> .
C	<i>n</i>	Cancella il testo dal cursore alla fine della riga e attiva la <i>modalità input</i> .
cc , S	<i>n</i>	Cancellano un'intera riga e attivano la <i>modalità input</i> .

x	<i>n</i>	Cancella il carattere in corrispondenza del cursore. Cancella il testo selezionato.
X	<i>n</i>	Cancella il carattere prima del cursore.
d	<i>n</i>	Cancella il testo indicato dal comando di spostamento specifico immediatamente seguente (per esempio: db , dB , d- , d(, d{ , dH , dIG). Cancella il testo selezionato.
D	<i>n</i>	Cancella il testo dal cursore alla fine della riga.
dd	<i>n</i>	Cancella un'intera riga.

P		Aggiunge (incolla) il testo (copiato o tagliato) prima del cursore.
p		Aggiunge (incolla) il testo (copiato o tagliato) dopo il cursore.

U		Rende maiuscolo il testo selezionato.
u		Rende minuscolo il testo selezionato.

u	<i>n</i>	Annulla una modifica (ripetuto, annulla più modifiche).
Ctrl-r	<i>n</i>	Ripristina una modifica annullata (ripetuto, ripristina più modifiche).

I comandi che agiscono nella *modalità input* (**R**) o che attivano la *modalità input* terminano con **Esc**.

I comandi di cancellazione sono anche comandi di taglio: il testo tagliato viene memorizzato per poter essere aggiunto (incollato) altrove.

Il comando **u** dopo la selezione differisce dal comando **u** da solo.

3.5 Ricerca

/stringa	Esegue la ricerca in avanti nel file.
/	Ripete la ricerca in avanti nel file.
?stringa	Esegue la ricerca a ritroso nel file.
?	Ripete la ricerca a ritroso nel file.
n	Ripete la ricerca nella direzione corrente.
N	Ripete la ricerca nella direzione opposta.

I comandi di ricerca (di una stringa o parola o frase) sono visualizzati nella riga di stato e devono essere confermati con **Invio**.

3.6 Vari

.	<i>n</i>	Ripete l'ultimo comando.
Ctrl-g		Visualizza informazioni sul file e sulla riga corrente.

3.7 Uscita

ZZ Consente di uscire da *Vim*, salvando le modifiche.

4 I comandi di *Ex*

I comandi di *Ex* sono preceduti da « : » e devono essere confermati con **Invio**. Eseguito un comando nella *modalità Ex*, si torna automaticamente alla *modalità Vi*.

4.1 Apertura di un altro file

:e <i>secondo-file</i>	Apre un altro file, che diventa il file corrente.
:e! <i>secondo-file</i>	Apre un altro file, annullando le modifiche.
:e#	Apre il file alternativo.
:e!#	Apre il file alternativo, annullando le modifiche nel file corrente.

L'apertura di un altro file consente il trasferimento (per copia o taglio) di parti di file da un file all'altro.

4.2 Impostazione delle opzioni

:se	Mostra le opzioni impostate dall'utente.
:se all	Mostra tutte le opzioni disponibili nelle loro impostazioni predefinite.
:se opzione?	Mostra l'impostazione o il valore di un'opzione.

:se opzione	Abilita un'opzione, limitatamente alla sessione.
:se noopzione	Disabilita un'opzione, limitatamente alla sessione.
:se opzione=valore	Attribuisce un valore ad un'opzione, limitatamente alla sessione.

4.3 Vari

:h Visualizza un testo di aiuto.

4.4 Salvataggio

:w	Salva il testo nel file già nominato.
:w nuovo-file	Salva il testo e nomina il nuovo file non ancora nominato.
:w secondo-file	Copia e salva il testo in un secondo file, senza perdere il file originale.

4.5 Uscita

:bd	Consente di uscire dal buffer, se non ci sono modifiche da salvare.
:bd!	Consente di uscire dal buffer, annullando le modifiche.

:q	Consente di uscire da <i>Vim</i> , se non ci sono modifiche da salvare.
:q!	Consente di uscire da <i>Vim</i> , annullando le modifiche.
:wq, :x	Consentono di uscire da <i>Vim</i> , salvando le modifiche.

Riferimenti bibliografici

[1] AA.VV. *help.txt*.

:h
file:///usr/share/vim/vim73/doc/help.txt

[2] AA.VV. Documentazione varia.
file:///usr/share/vim/vim73/doc/...

Indirizzi

Ex, Vi <http://ex-vi.sourceforge.net/>
Vim <http://www.vim.org/>



PGI – Piccole Guide d’Informatica

<http://lgcrsll.altervista.org/pgi/index.html>

LICENZA PGI – PICCOLE GUIDE D’INFORMATICA

Le PGI – PICCOLE GUIDE D’INFORMATICA, o più brevemente PGI, sebbene siano redatte con grande accuratezza, sono sprovviste di qualsiasi forma di garanzia: è sempre opportuno procedere a verifiche.

Tutte le PGI possono essere riprodotte e/o distribuite totalmente o parzialmente in piena libertà. La ridistribuzione commerciale delle PGI non è vietata, ma neanche incoraggiata.

Tutte le PGI possono essere modificate totalmente o parzialmente in piena libertà. Le modifiche fatte in accordo con l’autore originale rendono l’autore delle modifiche coautore. Le modifiche fatte senza l’accordo con l’autore originale sono ammesse ma non incoraggiate: sarebbe auspicabile che, semplicemente in ossequio al galateo, riportassero il nome dell’autore originale e il nome del revisore.

Lavori derivati dalle PGI o comprendenti parti di PGI sono non solo ammessi ma incoraggiati: sarebbe auspicabile che, semplicemente in ossequio al galateo, riportassero i riferimenti alle PGI da cui derivano o che comprendono.

Tutte le PGI possono essere tradotte in piena libertà: sarebbe auspicabile che le traduzioni, semplicemente in ossequio al galateo, riportassero il nome dell’autore e il nome del traduttore.