

Vim, una rapida guida all'uso

Rattotti Matteo

20-06-04

Abstract

Vim è un editor di testo potentissimo e colmo di funzionalità, per imparare a usare completamente vim è necessario un po' di tempo e di pratica, questa rapida guida si propone di introdurre i pochi comandi per cominciare ad usare da subito vim in parte come settarlo. Questa guida è scritta in base al mio utilizzo di vim, verranno affrontati solo pochi e utili comandi, per ulteriori informazioni su vim esistono libri molto esaustivi e documentazione copiosa. In Specifico questa guida guarderà all'uso di vim e gvim sul MacOSX anche se in realtà puo' essere presa come una guida generica per l'uso di vim su tutti gli *nix.

1 Primi passi

1.1 Lanciare vim:

Cominciamo con aprire vim, puo' essere fatto in due modi, o aprendo il terminale, scrivere vim e battere un return, o fare click sull'icona di vim, in macosx facendo click sull'icona di vim una secondo volta non verrà aperta una seconda finestra, per far ciò esiste l'icona gvim, che ad ogni lancio apre una nuova istanza di vim, e puo' essere usato come drop target sul doc.

1.2 Dentro alla bestia:

Ok, siamo dentro, si vede una schermata con delle informazioni degli strani trattini sulla sinistra e scrivendo non accade nulla!! Infatti vim ha diverse modalit'a di utilizzo, quando viene lanciato di default si trova in modalità comando dove possono essere fatte molte operazioni sul testo, per cominciare

a scrivere basterà premere 'i' e si passerà al modo inserimento e da lì potrete scrivere quello che vi pare.

1.3 Le modalità di vim:

Come abbiamo appena capito vim dispone di più modalità, vediamo nello specifico quali sono e come accedervi, prima di tutto teniamo presente l'importantissimo tasto 'esc' che ci riporta in modalità comando, dalla quale possiamo passare in tutte le modalità. Ecco le tre modalità principali di vim:

1. - Comando: dai qui possiamo dare comandi specifici all'editor si ottiene premendo 'esc'
2. - Inserzione: si può inserire testo come un normale editor, dalla modalità comando ci si accede con 'i'
3. - Visuale: modalità in cui si può lavorare su porzioni di testo a blocchi, con le selezioni, sempre dal modo comando si attiva con 'v'

1.4 Qualche comando fondamentale:

Bene direte, ho lanciato vim, ho premuto 'i' ho scritto quello che dovevo, ma ora??? Giustamente è il momento di introdurre qualche comando base, per salvare, caricare un file e almeno uscire da vim ;-)

Ovviamente tutti i comandi vanno lanciati dalla modalità comando, altrimenti non farete altro che scrivere il comando nel testo, quindi premete 'esc' e provate i seguenti comandi:

- :w nomefile (w sta per write) Questo comando scrive il file con il nome che preferite, prima del nome ovviamente potete inserire il percorso dove lo volete salvare, da notare che vim con il tab fornisce l'autocompletamento dei percorsi, quindi non dovete sapere a memoria ogni cartella.
- :e nomefile (e sta per edit) Questo comando apre il file specificato, una volta aperto, per salvare le modifiche basta usare :w senza null'altro.
- :q (q per quit) Esce da vim

Qualche volta puo' capitare che vim vi segnali un errore, ad esempio se volete uscire senza salvare le modifiche, vim vi farà notare che uscendo perderete tutte le modifiche, e con :q non vi permetterà di uscire, oppure se create un file e cercate di scriverlo sopra uno già esistente, per forzare vim a eseguire il comando basta mettere un '!' dopo il comando, questo ignorerà i messaggi di errore e compirà puntualmente il suo dovere (es :w! ecc....).

1.5 Muoversi in vim:

Abbiamo visto come leggere, scrivere e uscire con vim, ora vediamo come muoverci più velocemente dentro al testo, nella modalità inserimento ci si muove con le freccette normalmente, ma se andiamo nella modalità comando si possono scoprire modi molto simpatici di spostarsi nel testo.

La sequenza di tasti sulla tastiera 'hjkl' ci permette di spostarci rispettivamente di un carattere a sinistra con 'h', in alto con 'k', in basso con 'j' e a destra con 'l', nulla di estasiante per ora, ma se davanti a una di queste lettere ponete un numero, invece di un carattere ci si sposta del numero di caratteri che avete scritto... Quindi '12l' ci fa spostare di 12 posizioni a destra, non male vero? Ci sono inoltre altri simpatici comandi che possiamo usare per spostarci:

- 'w' e 'b' ci spostano rispettivamente sul primo carattere della parola successiva o precedente
- 'e' e 'ge' come prima ma sull'ultimo carattere
- '\$' ci sposta a fine riga e '0' a inizio riga
- 'G' e 'gg' ci portano all'ultima e alla prima riga
- ':n' (n e' un numero) ci porta alla riga n
- 'zz' centra la riga su cui c'è il cursore, 'zt' la fa diventare la prima e 'zn' l'ultima

2 Editare un testo in modalità comando

2.1 Cancellare:

Vim ci permette di cancellare velocemente parti del testo, i seguenti comandi ci vengono spesso in aiuto:

- 'x' cancella il carattere sotto il cursore 'nx' cancella n caratteri da quello sotto il cursore verso dx
- 'dw' cancella una parola 'dnw' cancella n parole
- 'dd' cancella una riga 'n dd' cancella n righe
- 'D' cancella tutta la parte della riga a dx del cursore

Quando si cancella qualcosa, in realtà si è eseguito un taglia e il testo cancellato si trova in memoria pronto ad essere incollato ;-)

Se per caso ci sbagliamo a cancellare, sempre in modalità comando basta premere 'u' per fare un undo e tornare un passo indietro, (ctrl-r o mela-r per il redo).

2.2 Copia, incolla:

Come ho scritto primo il taglia è la semplice cancellazione, ora vedremo i comandi per copiare e incollare:

- 'y' copia il carattere sotto il cursore, possiamo però associare a y i comandi di prima e copiare una parola con 'yw' o 'nyw' copia n parole o 'y\$' copia fino alla fine della riga
- 'Y' o 'yy' copia tutta la riga
- 'p' incolla dopo il cursore e 'P' incolla prima del cursore

All'inizio potrebbe sembrare un po' macchinoso, ma dopo un po' di pratica vi accorgete come in realtà sia comodo e veloce.

2.3 Ricerche e sostituzioni:

Ovviamente in un editor cercare e far sostituzioni sono due fattori critici, vim ci permette di fare queste operazioni in modo molto completo, vediamo ora i comandi di ricerca:

- `[/parola]` cerca tutte le occorrenze di parola in avanti `?parola` ricerca a ritroso
- L'operatore `.` ci permette di cercare parole di cui o non conosciamo un pezzo o più parole che hanno un suffisso o un prefisso in comune es: `/ca.` troverà tutte le parole che cominciano con `'ca'`
- `*` e `#` cercano la parola sotto in cursore rispettivamente avanti e indietro

Passiamo ora alle sostituzioni:

- `:s/parola1/parola2` sostituisce la prima occorrenza di parola1 con parola due, per farlo su tutta una riga si deve aggiungere in `/g` alla fine
- `:s/parola1//` sostituisce parola1 con il vuoto
- `:s/$/parola1` aggiunge parola1 alla fine del testo e così per tutti gli operatori
- `:s/parola1/&parola2` concatena le due stringe
- Si possono fare sostituzioni in un range di righe semplicemente specificando la prima e l'ultima riga `:4,7s/parola1/parola2/g` sostituisce tutte le occorrenze da riga 4 a riga 7, per farlo in tutto il testo basta fare `:1,$/s/parola1/parola2/g`

Naturalmente non abbiamo che toccato la punta dell'iceberg sull'uso di vim, ma per cominciare questo dovrebbe essere sufficiente, come ultima cosa vediamo la preferenze di vim e come settarlo un po' meglio

3 Settare vim

Vim usa due file, `.vimrc` o `.gvimrc` nella vostra home, a seconda che usiate il vim da terminale o `gvim`, in questi file possono essere messe le vostre preferenze che vim legge al suo avvio.

3.1 Settare le opzioni:

Come si fanno a settare le proprie opzioni? Bhe prima cosa è necessario sapere cosa si vuole settare, poi il comando 'set' seguito dalle opzioni fa per noi, faccio alcuni esempi presi dal mio .gvimrc:

- 'set mousehide' questo fa sì che il cursore sparisca quando scrivo
- 'set cindent' questo più utile setta l'indentazione di default
- 'set expandtab' questo fa in modo che i tab siano trasformati in spazi
- questi tre comandi fanno sì che vim faccia dei file di backup 'set backup' 'set backupdir= /.vim/backup' 'set directory= /.vim/tmp' e che vengano messi nella cartella .vim/backup e .vim/temp i file temporanei

I comandi di vim che possono essere settati sono moltissimi, dentro vim un ':help' vi darà una lista di possibili comandi e opzioni, basta avere un po' di pazienza e di voglia di imparare =).

4 Fonti:

Cito qui le fonti da cui ho preso le informazioni:

www.google.com

www.vim.org

SIForge.org

Il mio sito www.rknet.it e la mia mail whamoo@rknet.it

Buon uso di Vim.