

Prima di iniziare: 2 brevi parole sugli editor...

Tutte le distribuzioni basate su Debian (Ubuntu/Knoppix/Mephis/DSL, ecc) utilizzano come editor predefinito “nano”, ma **ATTENZIONE**, “nano” non è l'editor standard delle distro Gnu/Linux, “nano” è comodo e per molti utenti “*dà assuefazione*”, quindi cercate di imparare ad usare l'editor più comune che è installato su “quasi tutte le distro”, ossia “VI” (VI sta per Visual editor) o per comodità la sua evoluzione “VIM” (alcune piccole eccezioni sono rappresentate dalle distro tipo FREESCO che stanno tutte su un floppy).

Vediamo i principali comandi [per un corretto utilizzo vi consiglio la lettura della man page di VI]:

- ✓ aprire un file con vi --> vi nome_file
- ✓ editare una riga --> vi posizionate sulla riga che volete editare e premete il tasto “i”, ora potete scrivere tutti i caratteri che volete. Tenete presente che il tasto [canc] non funziona in VI
- ✓ per cancellare un carattere o premete il tasto backspace dopo aver premuto il tasto “i”, oppure premete il tasto [esc] ed utilizzate il tasto “x” con la stessa modalità con cui generalmente utilizzate il tasto [canc]
- ✓ per uscire dal file vi sono varie possibilità, premete [esc], ora quello che scrivete comparirà in basso a sinistra, se volete salvare scrivete “:w”, mentre se volete uscire “:q” infine se volete salvare ed uscire “:wq”

I comandi

La sintassi di un comando (espressione non propriamente corretta che abbrevia l'espressione “una linea di comando”) è la seguente:

comando [opzione] [argomento]

dove i termini racchiusi tra parentesi quadre possono o meno essere specificati, in base a ciò che si vuole ottenere dal comando.

Esempio, provate a dare i seguenti comandi uno dopo l'altro:

ls

ls -lh

ls -lh .*

Vediamo ora alcuni comandi:

Partiamo con quelli che riguardano la struttura del filesystem, in particolare:

- ✓ **mkdir** che serve per creare una directory (mkdir conoscerlinux);
- ✓ **rmdir** che serve a rimuovere le directory (rmdir conoscerlinux) NB: la directory viene rimossa solamente se al suo interno non vi sono file;
- ✓ **cd** consente di cambiare directory (cd conoscerlinux oppure cd ..)
- ✓ **pwd** ci dice in che directory ci troviamo (pwd);

Esaminiamo ora quelli relativi ai file:

- ✓ **touch** crea un file vuoto, che successivamente noi potremo riempire (touch mio_nome);
- ✓ **rm** rimuove il file o la cartella che desideriamo [molto importanti le opzioni che gli passiamo] (rm mio_nome);
- ✓ **mv** consente di rinominare un file se lo si trasferisce all'interno della stessa directory, oppure consente di spostare il file stesso in una directory differente (mv mio_nome primo_file);
- ✓ **cat** stampa a video tutto il contenuto del file (cat mio_nome);
- ✓ **tail** mostra la parte terminale del file, indicativamente le ultime 10 righe ed è molto utile per analizzare i log (tail mio_nome);
- ✓ **ln** serve per creare i link ai file (o alle directory), ossia a mantenere un file con un determinato nome e dargli anche un sinonimo, oppure per far credere al sistema che un determinato file sia anche in un'altra directory (ln mio_nome mio_nome_link);
- ✓ **cp** consente di effettuare la copia di un file o di una directory (cp mio_nome mio_nome_copiato);
- ✓ **echo** stampa a video tutto quello che gli viene passato, ossia che sta dopo di esso (echo questo è quello che viene stampato);
- ✓ **grep** stampa le righe che corrispondono ad un modello, ossia filtra quello che gli viene passato in input e restituisce solamente quello che contiene ciò che gli è stato chiesto (ls | grep io);
- ✓ **more** è un comando “filtro”, che consente di visualizzare i file molto lunghi per pagine (du | more);
- ✓ **df** è il comando che permette di visualizzare l'ammontare di spazio libero delle varie partizioni (df come standard fornisce la misura in kilobyte, ma se gli passiamo il parametro -h, ossia df -h otteniamo il risultato in modo più leggibile, ossia in Megabyte o in Gigabyte);
- ✓ **du** è il comando che ci indica la quantità di spazio utilizzata su hard-disk dalla directory corrente (comprendente le sue sotto directory), stessa cosa di prima per il parametro -h (du);
- ✓ **tar** più che un comando, tar è un'utility di programmi che consente di comprimere e decomprimere i file secondo vari formati (rar, zip, tar.gz, tar.bz2, ecc...)

Piccola guida a VI

Vediamo una semplice sessione di editing:

- 1) vi mio_file (stai creando il file mio_file)
- 2) premi 'a' per entrare in modalita' input e digita 'questo e' un semplice file di testo'
- 3) ora premi ':' per entrare in modalita' riga comando e digita 'w!' (con questo comando si scrive il file e si esce da VI)

Principali comandi in modalita' comando

- w -> vai all'inizio della prossima parola
- b -> vai all'inizio della precedente parola
- e -> vai alla fine della prossima parola
- nl -> vai alla colonna 'n' della riga corrente
- \$ -> vai alla fine della riga corrente
- ^ -> vai al primo carattere della riga corrente
- G -> vai alla fine del file
- gg -> vai all'inizio del file
- Gn -> vai alla riga 'n' del file
- i -> modalita' inserimento (prima del cursore)
- a -> modalita' inserimento (dopo il cursore)
- o -> modalita' inserimento (sotto la riga corrente)
- x -> cancella un carattere
- nx -> cancella 'n' caratteri
- dd -> cancella l'intera riga
- ndd -> cancella 'n' righe
- yy -> copia la riga corrente
- p -> incolla la riga copiata
- Rmistringa -> sostituisce i prossimi caratteri con la stringa 'mistringa'
- /stringa -> trova la prossima stringa 'stringa'
- ?stringa -> trova la stringa 'stringa' all'indietro
- n -> ripete la ricerca nuovamente
- u -> annulla l'ultimo comando

Principali comandi in modalita' riga comando

- :w -> scrive il file
- :w nome_file -> scrive il file come 'nome_file'
- :q -> esce da vi
- :wq -> scrive ed esce dal file
- :q! -> esce dal file senza salvare
- :help -> apre il file di aiuto (:q per tornare ad editare il file)
- :help argomento -> apre il file di aiuto all'argomento specificato da 'argomento' (:q per tornare ad editare il file)
- :rs/stringa1/stringa2/opzioni -> sostituisce stringa1 con stringa2 all'interno del range 'r'; le opzioni possono essere: g (sostituisce tutto), c (chiede conferma prima di ogni sostituzione)
- :set number -> imposta la numerazione delle righe

Puoi imparare altri comandi (ce ne sono tantissimi) usando il comando :help. Certo, e' anche possibile usare il comando 'man vi'. Il miglior modo per imparare VI e' iniziare ad usarlo

Inoltre potrete trovare su http://www.volalibero.it/immagini/vim_manuale.pdf un'ottima guida a VI per principianti