

Basic vi Editor Commands

Jemmy Hu
SHARCNET
University of Waterloo

What is vi?

vi (visual editor) is the default editor that comes with the UNIX operating system. Alternate editors for UNIX environments include **nano** (formerly named **pico**) and **emacs** (Basic emacs Commands, <http://www.cs.colostate.edu/helpdocs/emacs.html>)

The UNIX vi editor is a full screen editor and has two modes of operation:

1. *Command mode* commands which cause action to be taken on the file, and
2. *Insert mode* in which entered text is inserted into the file.

In the command mode, every character typed is a command that does something to the text file being edited; a character typed in the command mode may even cause the vi editor to enter the insert mode. In the insert mode, every character typed is added to the text in the file; pressing the <Esc> (*Escape*) key turns off the Insert mode.

While there are a number of vi commands, just a handful of these are usually sufficient for beginning vi users.

Both UNIX and vi are **case-sensitive**. Be sure not to use a capital letter in place of a lowercase letter; the results will not be what you expect.

1) Invoking vi

vi file_name

you will see the screen with a column of tildes. The editor vi is now in so called *command mode*.

The screen looks like:

~

Exercise: create a test file using *vi test1* and type a few line characters.

2) Exit vi

Usually the new or modified file is saved when you leave vi. However, it is also possible to quit vi without saving the file.

Note: The cursor moves to bottom of screen whenever a colon (:) is typed. This type of command is completed by hitting the <Return> (or <Enter>) key.

*	:x<Return>	<i>quit vi, writing out modified file to file named in original invocation</i>
	:wq<Return>	<i>quit vi, writing out modified file to file named in original invocation</i>
	:q<Return>	<i>quit (or exit) vi</i>
*	:q!<Return>	<i>quit vi even though latest changes have not been saved for this vi call</i>

3) Inserting or Adding Text

The following commands allow you to insert and add text. Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.

* i	<i>insert text before cursor, until <Esc> hit</i>
* a	<i>append text after cursor, until <Esc> hit</i>
* o	<i>open and put text in a new line below current line, until <Esc> hit</i>

4) Cursor movements commands

You need to be in the command mode. If you do not now what mode is actual, press the button esc. This keystroke always turns the editor in the command mode. Then you can move along the screen if you keystroke the button:

Keystroke of	Doing
h	Cursor is moved one space to the left
j	Cursor is moved one line down
k	Cursor is moved one line up
l	Cursor is moved one space to the right

5) Changing text

Very useful is so called *replace mode* which enables overwrite the existing text.

Keystrokes	Doing
r	Replace one character over the cursor
R	Overwrite text until the next action (e.g. keystroke of <i>esc</i>)

6) Deleting Text

The following commands allow you to delete text.

* x	<i>delete single character under cursor</i>
* dd	<i>delete entire current line</i>

7) Cutting and Pasting Text

The following commands allow you to copy and paste text.

yy	<i>copy (yank, cut) the current line into the buffer</i>
p	<i>put (paste) the line(s) in the buffer into the text after the current line</i>

8) Screen Manipulation

The following commands allow the vi editor screen (or window) to move up or down several lines and to be refreshed.

In the table below, the symbol ^ before a letter means that the <Ctrl> key should be held down while the letter key is pressed.

^f	<i>move forward one screen</i>
^b	<i>move backward one screen</i>
^d	<i>move down (forward) one half screen</i>
^u	<i>move up (back) one half screen</i>

9) Searching Text

A common occurrence in text editing is to replace one word or phrase by another. To locate instances of particular sets of characters (or strings), use the following commands.

/string	<i>search forward for occurrence of string in text</i>
?string	<i>search backward for occurrence of string in text</i>
n	<i>move to next occurrence of search string</i>

10) Determining Line Numbers

Being able to determine the line number of the current line or the total number of lines in the file being edited is sometimes useful.

::=	<i>returns line number of current line at bottom of screen</i>
:=	<i>returns the total number of lines at bottom of screen</i>
^g	<i>provides the current line number, along with the total number of lines, in the file at the bottom of the screen</i>
:set number	<i>provides the line number for the whole file</i>

References:

Basic vi Commands, <http://www.cs.colostate.edu/helpdocs/vi.html>