“vi” made less difficult

by Jeff Offutt

The standard `UNIX` full screen editor is **vi** (for **vi**sual editor) This document describes a very small set of **vi** commands — nowhere near the full power of **vi**, but enough to do most operations. **vi** is an extremely powerful and equally confusing editor. Very few people know all of its capabilities. The way I learned was to start with a small set then expand my knowledge slowly through the help guide and by asking questions.

1 vi Modes

vi is always in one of three states, or modes: *command*, *insert*, or *last-line*.

Command This is the initial mode. In command mode, every key is a command. The only warning indicator **vi** gives you is a beep. If you get a beep, that means you have not goofed anything up. If you are unsure what mode you are in, press the escape key (<`esc`>) — that will always return you to command mode.

Insert This is the mode in which you add text to your file. Insert is entered by a variety of commands, a couple of which are described below. Pressing the escape key (<`esc`>) will end insert mode.

Last-Line Typing a “:” (colon) will place you into *last-line* mode (also sometimes called *command-line* mode). The cursor will hop to the bottom of your screen and you can enter various commands. An escape or carriage return will end *last-line* mode.

2 Commands

Some basic commands that are enough to perform most editing are described here. Your efficiency in using **vi** will increase dramatically if you learn new operations as soon as you are comfortable with these. Enter **vi** by typing “`vi` *filename*”.

Remember that no carriage return is needed to enter a command when you’re in command mode.

2.1 Positioning the Cursor

Note that the four character positioning keys are found next to each other. The way to use **vi** is to put your right hand down with your four fingers covering the keys “`hjkl`” and learn to instinctively press the correct finger to move the cursor in the direction you want to go.

j Moves your cursor one line down on your screen.

k Moves your cursor one line up on your screen.

h Moves your cursor one character to the left on your screen.

l Moves your cursor one character to the right on your screen.

CR (*carriage return*) Moves your cursor to the beginning of the next line.

^**F** This moves you one entire screen-full forward through the file. (Note that "^F" means `<ctrl>F`.)

^**B** This moves you one entire screen-full backwards through the file. (Note that "^B" means `<ctrl>B`.)

G Moves the cursor to the last line in the file.

1G Moves the cursor to the first line in the file. (*n*G moves the cursor to the *n*th line in the file.)

^**G** Displays what line number you are on. (Note that "^G" means `<ctrl>G`.)

^**L** Correctly redisplays your file if you have a "messed up" screen. (Note that "^L" means `<ctrl>L`.)

2.2 Operations on Text

i Insert before the character. This places you into insert mode before the current position. `<esc>` will return you to command mode.

o Open a new line below the current line. Once the line is created, you are in insert mode.

d Delete. This key is used in conjunction with another key to specify how much to delete. "`dd`" deletes a line, "`dw`" deletes a word, and "`d$`" deletes to the end of the current line. A number entered in front of this key causes extra deletions to take place; for example, "`4dd`" deletes four lines.

x Delete one character.

p Pastes back the last thing you deleted. This is very useful for moving text around. Simply delete the text you want moved, move the cursor to where you want the text to be inserted, and press the "`p`" key to paste the text after the cursor. The "`P`" key does the same thing except that it pastes the text before the cursor.

u The most important command! "`u`" undos the most recent command when you goof up.

2.3 Saving Files and Terminating vi

:w This *last-line* command writes your text back to the file. Note the ":", which makes it a *last-line* command. If you supply a file name, **vi** will write to that file, else it will write to the file that was supplied when you called **vi**.

:wq This combination *last-line* command writes your text back to the file and quits the editor.
(The "**:x**" command does the same thing.)

:q! This *last-line* command quits the editor *without saving any changes you may have made.* (This command is used only when you've made a mess of your file and *don't* want to save it in its current state.)