

# Linux vi Editor

Vi was originally developed by William Joy at Berkeley University and first officially included in AT&T System 5 Unix. It started out as a line-oriented editor for dumb terminals. The Vi editor, or Vi iMproved (VIM) is [an enhancement by Bram Moolenaar](#); it's the version that people use today.

## Getting Started

To start Vi, open a terminal or console and simply type "vi" (without the quotation marks) followed by the name of any existing file or a new file you want to create.

The vi editor has three modes, command mode, insert mode and command line mode.

1. **Command mode:** letters or sequence of letters interactively command vi. Commands are case sensitive. The ESC key can end a command.
2. **Insert mode:** Text is inserted. The ESC key ends insert mode and returns you to command mode. One can enter insert mode with the "i" (insert), "a" (insert after), "A" (insert at end of line), "o" (open new line after current line) or "O" (Open line above current line) commands.
3. **Command line mode:** One enters this mode by typing ":" which puts the command line entry at the foot of the screen.

Vi works in two main modes, one for editing text and the other for giving commands. To switch between the two modes you use the I and Esc keys. The program opens in the Command mode, which is used for cursor movements, delete, cut, copy, paste, and saving changes.

The Insert mode is what you'll work in most of the time. You use it to make changes in an open file. Enter the Insert mode by pressing the I key. Newer Vi versions will display the word "INSERT" on the bottom line while you're in Insert mode.

Press the Esc key to switch Vi back to Command mode. As soon as you hit the Esc key the text "INSERT" on the bottom line disappears.

You save your changes to the open file from the Command mode. Press Shift-ZZ to save.

If you make a mistake when saving a file, such as pressing Ctrl-ZZ or closing Vi before saving the file, you'll end up with a swap file (akin to a DOS/Windows temp file) in addition to the original file. Usually the swap file will have the .swp extension. The original file will not contain the recent changes you made; attempting to reopen it will result in an error message.

The swap file is not readable but can be recovered by typing a command like this at the \$ prompt and pressing Enter:

```
vi -r {your file name}
```

In some extreme cases, recovery is not possible. But in most cases, such as closing Vi before saving, a system crash, or a power failure, recovery works very well. After you recover, you must manually delete the swap file using a command like this at the \$ prompt:

```
rm .{your file name}.swp
```

### **Step Vi Program**

Once you get used to it, using Vi is as easy as eating French fries. Try this exercise to get started. (Please follow all the instructions to the end.)

**1.** Make a new file called "tessst" by opening a console and typing this line after the \$ prompt and press Enter:

```
vi tessst
```

**2.** You'll get an empty console screen since Vi will start with the empty new file. Remember, Vi always starts Command mode, so press the I key to enter the Insert mode. If you're ever not sure whether you're in Insert mode, you can always just hit I again.

**3.** Next, type this line:

```
"The quick brown fox."
```

**4.** Press the Esc key to return to the Command mode.

**5.** Save the file by holding down the Shift key and pressing the Z key twice: ZZ.

**6.** Vi should close and you should see your \$ prompt back in the console.

**7.** Check to see if your additions to the file were correctly saved. At the \$ prompt, type:

```
cat tsssst
```

The Cat command lists the file's contents. You should see:

"The quick brown fox" line, exactly as you typed it earlier.

**8.** Next, reopen the file you created. At the \$ prompt, type this and press Enter:

```
vi tsssst
```

This time Vi opens up to the text in your file, not a blank screen.

**9.** Put Vi back in Insert mode by pressing I and add another line of text:

"The sleepy dog did not notice the clever fox."

**10.** Now save the file again by pressing Esc to enter the Command mode and typing Shift-ZZ.

**11.** Have a look at your changes again by using the Cat command at the \$ prompt:

```
cat tsssst
```

**12.** Repeat the entire process again, adding yet another line to your file. When you've done that, review the file contents of your home directory by typing this at the \$ prompt and pressing Enter:

```
ls -a
```

There should be only one file called "tsssst." If there's more than one, no matter what the extension, you made a mistake somewhere along the way. Delete all the tsssst files repeat the steps from Step 1. This command will delete swap files you may have created:

```
rm .tsssst.swp
```

If you made no mistakes, go have some French fries.

### Partial list of interactive commands:

#### Cursor movement:

Keystrokes	Action
h/j/k/l	Move cursor left/down/up/right
-/+	Move cursor down/up in first column
ctrl-d	Scroll down one half of a page
ctrl-u	Scroll up one half of a page
ctrl-f	Scroll forward one page
ctrl-b	Scroll back one page
M (shift-h)	Move cursor to middle of page
H	Move cursor to top of page
L	Move cursor to bottom of page
w	Move cursor a word at a time
5w	Move cursor ahead 5 words
b	Move cursor back a word at a time
B	Move cursor back a word at a time
5b	Move cursor back 5 words
e	Move cursor to end of word
5e	Move cursor ahead to the end of the 5th word
0 (zero)	Move cursor to beginning of line
\$	Move cursor to end of line
G	Move cursor to end of file
%	Move cursor to the matching bracket. Place cursor on {}[]() and type "%".
.'	Move cursor to previously modified line.
'a	Move cursor to line mark "a" generated by marking with keystroke "ma"
'A	Move cursor to line mark "a" (global between buffers) generated by marking with keystroke "mA"
]'	Move cursor to next lower case mark.
['	Move cursor to previous lower case mark.

### Editing commands:

Keystrokes	Action
i	Insert at cursor
a	Append after cursor
A	Append at end of line
ESC	Terminate insert mode
u	Undo last change
U	Undo all changes to entire line
o	Open a new line
dd 3dd	Delete line Delete 3 lines.
D	Delete contents of line after cursor
dw 4dw	Delete word Delete 4 words
cw	Change word
x	Delete character at cursor
r	Replace character
R	Overwrite characters from cursor onward
~	Change case of individual character
ctrl-a ctrl-x	Increment number under the cursor. Decrement number under the cursor.
/search_string{CR}	Search for <i>search_string</i>
?search_string{CR}	Search backwards (up in file) for <i>search_string</i>
^<search_string>{CR}	Search for <i>search_word</i> Ex: ^<s> Search for variable "s" but ignore declaration "string" or words containing "s". This will find "string s;", "s = fn(x);", "x = fn(s);", etc
n	Find next occurrence of search_word
N	Find previous occurrence of search_word
.	repeat last command action.

### Terminate session:

- Use command: ZZ  
Save changes and quit.

- Use command line: ":wq"  
Save (write) changes and quit.
- Use command line: ":w"  
Save (write) changes without quitting.
- Use command line: ":q!"  
Ignore changes and quit. No changes from last write will be saved.
- Use command line: ":qa"  
Quit all files opened.

# The VI Editor

<b>ENTER VI</b> vi file	<b>DELETIONS</b> Char under cursor      x Char before cursor    X Line                      dd To end of line          D To start of next word   dw
<b>CURSOR MOVEMENT</b> <b>LOCAL</b> Right one char        ->, l, (sp) Left one char        <-,h, (bs) Up one line          ▲, k, ctrl-p Down one line        ▼, j, ctrl-n <b>PAGE</b> Forward a page        ctrl-f Back a page           ctrl-b Forward a half page   ctrl-d Back a half page      ctrl-u <b>SCREEN</b> Top left              H Bottom left           L Middle line           M <b>LINE</b> Left margin           0 End                    \$ Next line             + Prior line             - N-th line             nG, :n Last line              :\$ <b>WORD</b> Next word             w Prior word            b	<b>TEXT MODE</b> Append after cursor    a Insert before cursor   i Append to end of line   A Insert at start of line   I Open next line        o Open prior line        O
<b>SEARCH/SUBSTITUTE</b> <b>SEARCH</b> Search forward        /string Search backward      ?string Repeat last search    n Reverse last search   N <b>SUBSTITUTE</b> On current line        :s/str1/str2/ All occurrences           :s/str1/str2/g Range of lines        :l1,l2s/str1/str2/	<b>TEXT REPLACEMENT</b> Char under cursor     r Overwrite chars        R . . . <esc> <b>SAVE/EXIT</b> Save                    :w Quit (no changes)     :q Quit (changes)        :q! Save & quit            :wq, ZZ Write to a file         :w file Overwrite to a file    :w! file <b>TEXT MOVEMENT</b> Yank lines             [cnt]Y, [cnt]yy Paste after cursor      p Paste before cursor    P Insert file             :r file <b>MISCELLANEOUS</b> Undo changes          u