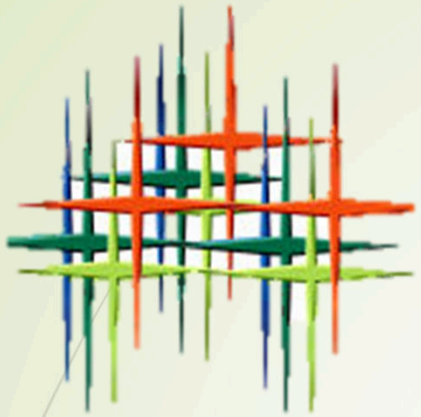


ABC IT Education

WE'LL TAKE YOU FROM ZERO TO HERO IN A SNAP

Linux Systems Administration

Editing Files in Vi, Vim, Emacs & Graphical Editors

vi and vim Editors

- Today we will learn how to use the 'vi' editor.
- Earlier we saw the ‘nano’ editor and it is a great editor for simple edits.
- ‘vi’ and ‘emacs’ on the other hand have more advanced and powerful features not available with ‘nano’
- Also, ‘vi’ and ‘emacs’ are not very intuitive and as easy to learn as ‘nano’ so learning ‘vi’ will definitely be a little harder than nano and will require a bit of a time investment to become proficient using them.
- To use the vi editor, simply type vi space and the file name that you want to edit. **$ vi <file name>**
- As an alternative, you can also use **vim <file name>**, where vim stands for vi improved.

- ‘vim’ is compatible with all the ‘vi’ commands but it has some additional features like syntax, highlighting and the ability to edit files over the network.
- Note that on many Linux distributions vi actually invokes vim or improved vi.
- To view a file but not make any changes to its contents, we the view command. **$ view <file name>**
- ‘vi’ is always available on a Linux system.
- Once you learn the key mappings for vi, you can apply them to other commands like man or more or less or view or you can even use vi mode for editing shell commands.
- ‘vi’ has the concept of modes, so you are always working in one of three modes - command mode, insert mode, or line mode.

- When vi starts, you are placed in the command mode, and to get back to command mode from any of the other modes, just hit the escape key.
- When you type letters while in command mode they are written into the file but they are interpreted as commands.
- The command mode allows you to navigate about the file, perform searches, delete text, copy text and paste text.
- Here are some of the commonly used key bindings for navigation and command mode.
- 'k' goes up one line, 'j' down, 'h' left, 'l' right one character. To move right one word, 'w', 'b' moves back one word or left one word. The '^' symbol takes you to the beginning of the line while the '$' symbol takes you to the end of the line.

- To insert text in a file, enter vi's insert mode and this is done by pressing 'i', 'I', 'a', or 'A'.

Vi Insert Mode

i	Insert at the cursor position.
I	Insert at the beginning of the line.
a	Append after the cursor position.
A	Append at the end of the line.

- To use vi's line mode, begin the command with colon.
- To write the file or save the file, type ':w', enter.
- To force a file to be saved, type ':w!'.
- To quit, ':q', to force quit, ':q!'.
- To save the file and quit, or write and quit, ':wq!'.
- And ':x' is the same as ':wq'

- Line mode can be used for navigation to, if you want to go to the line 15 in the file, type ':15' enter.
- To go to the last line of the file, type ':$'.
- To see line numbers, type ':set nu'.
- And to turn off line numbering, ':set nonu'.
- For help, type ':help' and the command you want help on.
- Again the modes in vi or command mode which you can enter by hitting the escape key. Insert mode by using 'i or a', and line mode by beginning commands with ':' character.
- You can repeat commands in vi by preceding them with a number, for instance, if you would like to move the cursor up 5 lines, simply type '5k'.
- To insert a piece of text 80 times, type '80i' and start entering the text.

- Once you're done, you hit the escape key to return to command mode and the text you type will be repeated 80 times.
- In command mode you can use 'x' to delete a character, 'dw' to delete a word, 'dd' to delete a line and 'D' to delete the remaining text on the line.
- To replace text, type 'r' for just one character, 'cw' to change an entire word, 'cc' to change in entire line, 'c$' to change the text from the current position to the end of the line, you can also use 'C' for that.
- The '~' command reverses the case of the character.
- To yank or copy the current line, type 'yy'
- To yank a position type 'y' and a position character.
- For instance, to yank a word, type 'yw'

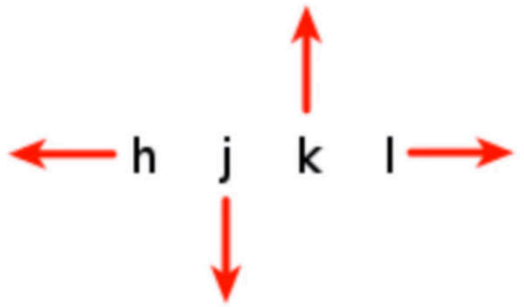
- If we want to repeat this, you could yank 3 words with 'y3w'.
- 'p' will paste the most recently deleted or yanked text.
- The undo command in vi is 'u' and the redo command is 'ctrl-R'.
- To start a forward search, type '/' and a search pattern and hit enter.
- To go to the next match, type 'n`, previous match, type 'N'.
- To start a reverse search, use the '?' followed by a search pattern and then hit enter.

vi Commands Summary

The Vi Editor

`vi [file]`	Edit file.
`vim [file]`	Same as vi, but more features.
`view [file]`	Starts vim in read-only mode.

Vi Navigation Keys



Vi Command Mode and Navigation

k	Up one line.
j	Down one line.
h	Left one character.
l	Right one character.
w	Right one word.
b	Left one word.
^	Go to the beginning of the line.
$	Go to the end of the line.

Vi Insert Mode

i	Insert at the cursor position.
I	Insert at the beginning of the line.
a	Append after the cursor position.
A	Append at the end of the line.

vi Commands Summary

Vi Line Mode

:w	Writes (saves) the file.
:w!	Forces the file to be saved.
:q	Quit.
:q!	Quit without saving changes.
:wq!	Write and quit.
:x	Same as :wq.

Vi Modes

Mode	Key
Command	Esc
Insert	i I a A
Line	:

Vi Line Mode

:n	Positions the cursor at line n.
:$	Positions the cursor on the last line.
:set nu	Turn on line numbering.
:set nonu	Turn off line numbering.
:help [subcommand]	Get help.

vi Commands Summary

Vi - Repeating Commands

- Repeat a command by preceding it with a number.
  - 5k = Move up a line 5 times
  - 80i<Text><ESC> = Insert <Text> 80 times
  - 80i_<Esc> = Insert 80 "_" characters

Vi - Deleting Text

x	Delete a character.
dw	Delete a word.
dd	Delete a line.
D	Delete from the current position.

Vi - Changing Text

r	Replace the current character.
cw	Change the current word.
cc	Change the current line.
c$	Change the text from the current position.
C	Same as c$.
~	Reverses the case of a character.

Vi - Copying and Pasting

yy	Yank (copy) the current line.
y<position>	Yank the <position>.
p	Paste the most recent deleted or yanked text.

Vi - Undo / Redo

u	Undo
Ctrl-R	Redo

Vi - Searching

/<pattern>	Start a forward search.
?<pattern>	Start a reverse search.

Using vi to edit a file

- Download vipractice.txt from the classnotes section.
- Open the file using vi, **$ vi vipractice.txt**
- You'll notice at the bottom of the screen, you see a series of tildes. They indicate the lines that are beyond the file.
- Remember ‘vi’ starts out in command mode so we can use the 'j' key to navigate down, 'k' goes up, 'l' to the right, 'h' to left.
- '^' symbol goes to the beginning of the line
- '$' sign goes to the end of the line.
- To enter insert mode, use 'i', to insert text at the current position.
- Always remember that **escape** will always take us back to command mode.

Using vi to edit a file

- To insert text at the very beginning of the line, 'shift i' or '**I**'.
- To append just use '**a**‘, to append to the end of the line is 'shift+a‘ or ‘**A**’
- To delete text, use the '**x**' to delete a character.
- To delete a word, use '**dw**‘, and '**dd**' will delete a line.
- '**D**' deletes from the current position to the end of the line.
- To replace text, we can use '**r**‘, will replace one character.
- Can replace an entire word with '**cw**' for change word.
- Escape ‘**esc**’ to get back to command mode.
- '**cc**' to change the entire line.
- And the '**~**' changes the case, so if it's a lowercase, it will become uppercase.

Using vi to edit a file

- To yank a line, '**yy**'
- To paste, '**p**'.
- To do a forward search, type the '**/**'
- and lowercase '**n**' to move to the next item.
- And uppercase '**N**' to move to the previous item.
- To do a backward search, start with a question mark, '**?**'.
- And to write changes to a file, we can use '**w**'.
- To quit, we can type '**q**', now that there are saved changes, it won't let us stop.
- So we'll just use '**q**' with an exclamation mark '**q!**'to force our quit.

Using vi to edit a file

- Reopen the file **$ vi vipractice.txt**
- Let's turn on line number, it was 'set nu'.
- And we can go to line 10 by typing ':10‘, or line 5, ':5' enter.
- And we'll quit with 'q'.

- There's also a vi tutor (**vimtutor**) that's very nice, and follow along and it will take you through some of the commands and features of vi.

emacs Editor

- Emacs like vi is a powerful editor, and some people prefer it over vi.
- Since bot editors are powerful, it is important to experiment with both, then choose the one you feel more comfortable with.
- To edit a file with emacs, run emacs followed by a space and the name of the file you wish to edit.
- $ emacs <filename>
- The emacs documentation shows a 'c-' or `m- character, that means to hold down the control key for `c-' or the meta or alt key for `m-' while pressing that character.
- As an example, 'c-h` means to hold down the control key while pressing the 'h' key.

- Some commands have more than 2 letters, like the case of 'c-h t‘ that means to hold down the control key while pressing the 'h' key, release the control key and then type the letter 't'.
- As indicated above, the 'm-' character, means that you hold down the meta key which is the 'alt' key.
- The 'esc' key can also be used in the place of the 'alt' key.
- For example, 'm-f' translates to holding down the alt key and pressing 'f', or pressing and releasing 'esc', followed by the 'f' key.
- In some instances, the ‘alt’ key is not available, intercepted by your terminal program, so you will need to use the 'esc' key as your meta key
- Always use the 'esc' key for the meta key, to simplify things.

emacs Commands

- 'C-h’ - Help.
- 'C-x' then 'C-c‘ - Exit emacs.
- 'C-x’ then ‘C-s‘ - Saves the file,
- 'C-h t' - Starts built-in emacs tutorial.
- 'C-h k' then a key - Describe that key.
- 'C-h k' can be used to get help on a specific key command or key combination.
- 'C-p‘ - Go to previous line,
- 'C-n‘ - Go to the next line.
- 'C-b’ - Go backward one character.
- 'C-f’ - Go forward one character.
- 'M-f’ - Move you forward one word

- 'M-b' - Take you back one word.
- ‘C-a‘ - Take you to the beginning of the line.
- ‘C-e' - Take you to the end of a line.
- 'M-<' - Take you to the beginning of the file
- 'M->’ - Take you to the end of the file.
- ‘C-d’ - To delete a character
- 'M-d’ - To delete a word, its.
- ‘C-k' - Kill the rest of the current line, from cursor
- To kill (delete) an entire line, position the cursor at the beginning of the line then type ‘C-k’.
- If cursor is not at the beginning of the line, the following two command in succession will kill the line ‘C-a’ and ‘C-k’

- ‘C-y’ - to yank or paste the previously killed text.

  Note in **emacs** yank = paste and in **vi** yank = copy
- ‘C-x u' - To undo
- ‘C-s' - To start a forward search, then type text you are searching for, followed by ‘C-s' again to move to the next occurrence of the text and finally enter when done.
- ‘C-r' - To start a reverse search.
- emacs allows for repetition of commands, example to delete 5 characters, type ‘C- u 5 C-d'.

emacs commands summary

Emacs

`emacs [file]`	Edit file.
`C-<char>`	Ctrl while pressing <char>.
`M-<char>`	"Meta" key (alt key) while pressing <char>.
`M-<char>`	Esc, then type <char>

Emacs Commands

`C-h`	Help
`C-x C-c`	Exit
`C-x C-s`	Save the file
`C-h t`	Built-in tutorial
`C-h k <key>`	Describe key

emacs commands summary

Emacs Navigation

C-p	Previous line.
C-n	Next line.
C-b	Backward one character.
C-f	Forward one character.
M-f	Forward one word.
M-b	Backward one word.

Emacs Navigation

C-a	Go to the beginning of the line.
C-e	Go to the end of the line.
M-<	Go to the beginning of the file.
M->	Go to the end of the file.

emacs commands summary

Emacs - Deleting Text

C-d	Delete a character.
M-d	Delete a word.

Emacs - Searching

C-s	Start a forward search.
C-r	Start a reverse search.

Emacs - Copying, Pasting, and Undo

C-k	Kill (cut)
C-y	Yank (paste)
C-x u	Undo

Emacs - Repeating Commands

C-u N <command>	Repeat <command> N times.

Using emacs to edit a file

- Open the file using emacs, **$ emacs vipractice.txt**
- To navigate, you can use ‘C-n' to go the next line, ‘C-f' will go forward one character, ‘C-p' will go previous line, ‘C-b' will go back a character.
- To go to the beginning of the line, type ‘C-a’ and ‘C-e’ to go to the end of the line.
- To enter text, just start typing where you are or go to the location where you want to enter the text and just type.
- Let's delete a character with ‘C-d'.
- Delete a word, type 'M-d‘, can use 'alt-d‘ or 'esc-d'.
- Kill the current line, type 'ctrl-k'.
- Move to the bottom of the file and yank (paste) the yanked line. Type 'M->’ then ‘C-y’

- Search for ‘This’, type ‘C-s’ then This, then ‘C-s’, then ‘enter’ when you are done with the search.
- For a reverse search of ‘This’, type ‘C-r’ then This, then ‘C-r’ to find the previous one and so on, and finally ‘enter’ to finish
- Type, ‘C-x C-c‘ to exit, no changes saved.
- To save the changes before exiting, type ‘C-x C-s‘, then ‘C-x C-c’
- Now that we have seen emacs, I still recommend you take time to check out the emacs tutorial, type ‘C-h t’ in emacs.

Graphical Editors

- Apart from Nano, Vi, Vim and Emacs, there are other editors, mostly graphical, available to you in Linux.
- When running Linux as a desktop operating system you can use one of any of the graphical editors available.
- Here are some graphical editors available to you in Linux graphical Emacs, 'gedit' which is the default text editor for the Gnome desktop environment.
- ‘gedit’ is very similar to notepad in Windows
- 'gvim' is the graphical version of vim.
- 'kedit‘, which is similar to 'gedit‘, but is the default editor for the KDE desktop environment.
- If looking for a Microsoft Word replacement, check AbiWord or LibreOffice, which is a complete office suite that contains a spreadsheet, a database and presentation software.

- If you're looking for a source code editor to help in computer programming, look at an editor like Kate, but there are other options available like Geany, or Gedit or even Sublime Text.