UNIX/LINUX - THE VI EDITOR TUTORIAL

In this chapter, we will understand how the vi Editor works in Unix. There are many ways to edit files in Unix. Editing files using the screen-oriented text editor **vi** is one of the best ways. This editor enables you to edit lines in context with other lines in the file.

An improved version of the vi editor which is called the **VIM** has also been made available now. Here, VIM stands for **Vi IM**proved.

vi is generally considered the de facto standard in Unix editors because –

- It's usually available on all the flavors of Unix system.
- Its implementations are very similar across the board.
- It requires very few resources.
- It is more user-friendly than other editors such as the **ed** or the **ex**.

You can use the **vi** editor to edit an existing file or to create a new file from scratch. You can also use this editor to just read a text file.

Starting the vi Editor

The following table lists out the basic commands to use the vi editor –

Sr.No.	Command & Description
1	**vi filename** Creates a new file if it already does not exist, otherwise opens an existing file.
2	**vi -R filename** Opens an existing file in the read-only mode.
3	**view filename** Opens an existing file in the read-only mode.

Following is an example to create a new file **testfile** if it already does not exist in the current working directory –

```
$vi testfile
```

The above command will generate the following output –

```
|
~
~
~
```

```
~
~
~
~
~
~
~
~
~
"testfile" [New File]
```

You will notice a **tilde** (~) on each line following the cursor. A tilde represents an unused line. If a line does not begin with a tilde and appears to be blank, there is a space, tab, newline, or some other non-viewable character present.

You now have one open file to start working on. Before proceeding further, let us understand a few important concepts.

Operation Modes

While working with the vi editor, we usually come across the following two modes –

- **Command mode –** This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.
- **Insert mode –** This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and placed in the file.

vi always starts in the **command mode**. To enter text, you must be in the insert mode for which simply type **i**. To come out of the insert mode, press the **Esc** key, which will take you back to the command mode.

Hint – If you are not sure which mode you are in, press the Esc key twice; this will take you to the command mode. You open a file using the vi editor. Start by typing some characters and then come to the command mode to understand the difference.

Getting Out of vi

The command to quit out of vi is **:q**. Once in the command mode, type colon, and 'q', followed by return. If your file has been modified in any way, the editor will warn you of this, and not let you quit. To ignore this message, the command to quit out of vi without saving is **:q!**. This lets you exit vi without saving any of the changes.

The command to save the contents of the editor is **:w**. You can combine the above command with the quit command, or use **:wq** and return.

The easiest way to **save your changes and exit vi** is with the ZZ command. When you are in the command mode, type **ZZ**. The **ZZ** command works the same way as the **:wq** command.

If you want to specify/state any particular name for the file, you can do so by specifying it after the **:w**. For example, if you wanted to save the file you were working on as another filename called **filename2**, you would type **:w filename2** and return.

Moving within a File

To move around within a file without affecting your text, you must be in the command mode (press Esc twice). The following table lists out a few commands you can use to move around one character at a time –

Sr.No.	Command & Description
1	**k** Moves the cursor up one line

2	**j** Moves the cursor down one line
3	**h** Moves the cursor to the left one character position
4	**l** Moves the cursor to the right one character position

The following points need to be considered to move within a file –

- vi is case-sensitive. You need to pay attention to capitalization when using the commands.
- Most commands in vi can be prefaced by the number of times you want the action to occur. For example, **2j** moves the cursor two lines down the cursor location.

There are many other ways to move within a file in vi. Remember that you must be in the command mode (**press Esc twice**). The following table lists out a few commands to move around the file –

Given below is the list of commands to move around the file.

Sr.No.	Command & Description
1	**0 or |** Positions the cursor at the beginning of a line
2	**$** Positions the cursor at the end of a line
3	**w** Positions the cursor to the next word
4	**b** Positions the cursor to the previous word
5	(Positions the cursor to the beginning of the current sentence

6	**)** Positions the cursor to the beginning of the next sentence
7	**E** Moves to the end of the blank delimited word
8	**{** Moves a paragraph back
9	**}** Moves a paragraph forward
10	**[[** Moves a section back
11	**]]** Moves a section forward
12	**n|** Moves to the column **n** in the current line
13	**1G** Moves to the first line of the file
14	**G** Moves to the last line of the file
15	**nG** Moves to the **nth** line of the file
16	**:n** Moves to the **nth** line of the file

17	**fc** Moves forward to **c**
18	**Fc** Moves back to **c**
19	**H** Moves to the top of the screen
20	**nH** Moves to the **nth** line from the top of the screen
21	**M** Moves to the middle of the screen
22	**L** Move to the bottom of the screen
23	**nL** Moves to the **nth** line from the bottom of the screen
24	**:x** Colon followed by a number would position the cursor on the line number represented by **x**

Control Commands

The following commands can be used with the Control Key to performs functions as given in the table below –

Given below is the list of control commands.

Sr.No.	Command & Description
1	**CTRL+d** Moves forward 1/2 screen

2	**CTRL+f** Moves forward one full screen
3	**CTRL+u** Moves backward 1/2 screen
4	**CTRL+b** Moves backward one full screen
5	**CTRL+e** Moves the screen up one line
6	**CTRL+y** Moves the screen down one line
7	**CTRL+u** Moves the screen up 1/2 page
8	**CTRL+d** Moves the screen down 1/2 page
9	**CTRL+b** Moves the screen up one page
10	**CTRL+f** Moves the screen down one page
11	**CTRL+I** Redraws the screen

Editing Files

To edit the file, you need to be in the insert mode. There are many ways to enter the insert mode from the command mode –

Sr.No.	Command & Description
1	**i** Inserts text before the current cursor location
2	**I** Inserts text at the beginning of the current line
3	**a** Inserts text after the current cursor location
4	**A** Inserts text at the end of the current line
5	**o** Creates a new line for text entry below the cursor location
6	**O** Creates a new line for text entry above the cursor location

Deleting Characters

Here is a list of important commands, which can be used to delete characters and lines in an open file –

Sr.No.	Command & Description
1	**x** Deletes the character under the cursor location
2	**X** Deletes the character before the cursor location
3	**dw** Deletes from the current cursor location to the next word

4	**d^** Deletes from the current cursor position to the beginning of the line
5	**d$** Deletes from the current cursor position to the end of the line
6	**D** Deletes from the cursor position to the end of the current line
7	**dd** Deletes the line the cursor is on

As mentioned above, most commands in vi can be prefaced by the number of times you want the action to occur. For example, **2x** deletes two characters under the cursor location and **2dd** deletes two lines the cursor is on.

It is recommended that the commands are practiced before we proceed further.

Change Commands

You also have the capability to change characters, words, or lines in vi without deleting them. Here are the relevant commands –

Sr.No.	Command & Description
1	**cc** Removes the contents of the line, leaving you in insert mode.
2	**cw** Changes the word the cursor is on from the cursor to the lowercase **w** end of the word.
3	**r** Replaces the character under the cursor. vi returns to the command mode after the replacement is entered.
4	**R** Overwrites multiple characters beginning with the character currently under the cursor. You must use **Esc** to stop the overwriting.

5	**s** Replaces the current character with the character you type. Afterward, you are left in the insert mode.
6	**S** Deletes the line the cursor is on and replaces it with the new text. After the new text is entered, vi remains in the insert mode.

Copy and Paste Commands

You can copy lines or words from one place and then you can paste them at another place using the following commands –

Sr.No.	Command & Description
1	**yy** Copies the current line.
2	**yw** Copies the current word from the character the lowercase w cursor is on, until the end of the word.
3	**p** Puts the copied text after the cursor.
4	**P** Puts the yanked text before the cursor.

Advanced Commands

There are some advanced commands that simplify day-to-day editing and allow for more efficient use of vi –

Given below is the list advanced commands.

Sr.No.	Command & Description
1	**J** Joins the current line with the next one. A count of j commands join many lines.
2	

	<< Shifts the current line to the left by one shift width.
3	**>>** Shifts the current line to the right by one shift width.
4	**~** Switches the case of the character under the cursor.
5	**^G** Press Ctrl and G keys at the same time to show the current filename and the status.
6	**U** Restores the current line to the state it was in before the cursor entered the line.
7	**u** This helps undo the last change that was done in the file. Typing 'u' again will re-do the change.
8	**J** Joins the current line with the next one. A count joins that many lines.
9	**:f** Displays the current position in the file in % and the file name, the total number of file.
10	**:f filename** Renames the current file to filename.
11	**:w filename** Writes to file filename.
12	**:e filename** Opens another file with filename.
13	

	:cd dirname Changes the current working directory to dirname.
14	**:e #** Toggles between two open files.
15	**:n** In case you open multiple files using vi, use **:n** to go to the next file in the series.
16	**:p** In case you open multiple files using vi, use **:p** to go to the previous file in the series.
17	**:N** In case you open multiple files using vi, use **:N** to go to the previous file in the series.
18	**:r file** Reads file and inserts it after the current line.
19	**:nr file** Reads file and inserts it after the line **n**.

Word and Character Searching

The vi editor has two kinds of searches: **string** and **character**. For a string search, the **/** and **?** commands are used. When you start these commands, the command just typed will be shown on the last line of the screen, where you type the particular string to look for.

These two commands differ only in the direction where the search takes place –

- The **/** command searches forwards (downwards) in the file.
- The **?** command searches backwards (upwards) in the file.

The **n** and **N** commands repeat the previous search command in the same or the opposite direction, respectively. Some characters have special meanings. These characters must be preceded by a backslash (\) to be included as part of the search expression.

Sr.No.	Character &Description
1	**^**

	Searches at the beginning of the line (Use at the beginning of a search expression).
2	. Matches a single character.
3	* Matches zero or more of the previous character.
4	**$** End of the line (Use at the end of the search expression).
5	[Starts a set of matching or non-matching expressions.
6	**<** This is put in an expression escaped with the backslash to find the ending or the beginning of a word.
7	**>** This helps see the '**<**' character description above.

The character search searches within one line to find a character entered after the command. The **f** and **F** commands search for a character on the current line only. **f** searches forwards and **F** searches backwards and the cursor moves to the position of the found character.

The **t** and **T** commands search for a character on the current line only, but for **t**, the cursor moves to the position before the character, and **T** searches the line backwards to the position after the character.

Set Commands

You can change the look and feel of your vi screen using the following **:set** commands. Once you are in the command mode, type **:set** followed by any of the following commands.

Sr.No.	Command & Description
1	**:set ic** Ignores the case when searching
2	**:set ai**

	Sets autoindent
3	**:set noai** Unsets autoindent
4	**:set nu** Displays lines with line numbers on the left side
5	**:set sw** Sets the width of a software tabstop. For example, you would set a shift width of 4 with this command — **:set sw = 4**
6	**:set ws** If *wrapscan* is set, and the word is not found at the bottom of the file, it will try searching for it at the beginning
7	**:set wm** If this option has a value greater than zero, the editor will automatically "word wrap". For example, to set the wrap margin to two characters, you would type this: **:set wm = 2**
8	**:set ro** Changes file type to "read only"
9	**:set term** Prints terminal type
10	**:set bf** Discards control characters from input

Running Commands

The vi has the capability to run commands from within the editor. To run a command, you only need to go to the command mode and type **:!** command.

For example, if you want to check whether a file exists before you try to save your file with that filename, you can type **:! ls** and you will see the output of **ls** on the screen.

You can press any key (or the command's escape sequence) to return to your vi session.

Replacing Text

The substitution command (**:s/**) enables you to quickly replace words or groups of words within your files. Following is the syntax to replace text –

```
:s/search/replace/g
```

The **g** stands for globally. The result of this command is that all occurrences on the cursor's line are changed.

Important Points to Note

The following points will add to your success with vi –

- You must be in command mode to use the commands. (Press Esc twice at any time to ensure that you are in command mode.)
- You must be careful with the commands. These are case-sensitive.
- You must be in insert mode to enter text.