VIM TEXT EDITOR ON LINUX UBUNTU

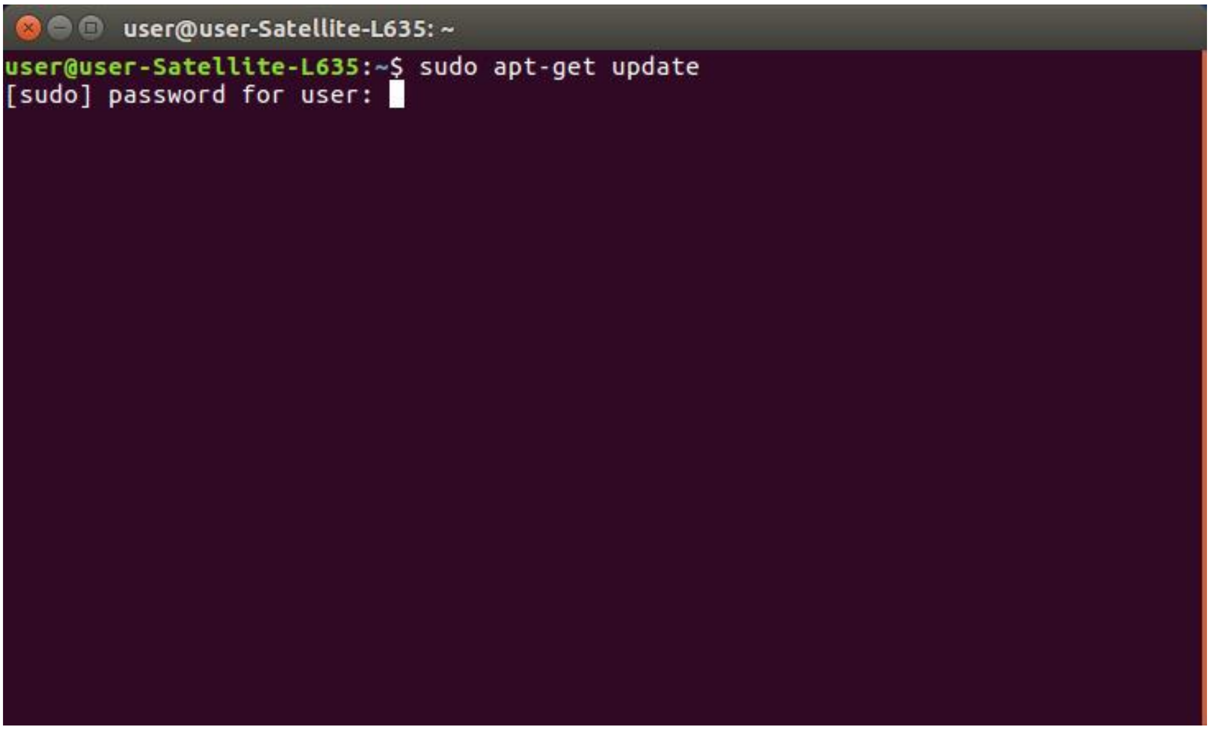
Vim is a very useful and helpful editor for creating and editing different types of files more efficiently. Many new features are added in this editor that makes it a powerful editor. Many plugins are developed by many coders for this editor to increase and configure its core functionalities. Some of them are Pathogen, Syntastic, indent guides, Fugitive, Git Gutter, etc. Git is a distributed version control system (DVCS) that helps the developers to manage the modified source codes over time. It is totally free to use. Using git command, the track changes and the revision history of the source codes can be easily traced. Git command works in the command line interface. The vim plugin named fugitive plugin is developed by Tim pope which is used to work with the git tool without terminating the editor. So, vim and git can work together by using the fugitive plugin.

INSTALLATION OF VIM

Step 1: First open the terminal application and then update package database using the apt command or apt-get command:

```
sudo apt-get update
```

Next enter the password for the linux.

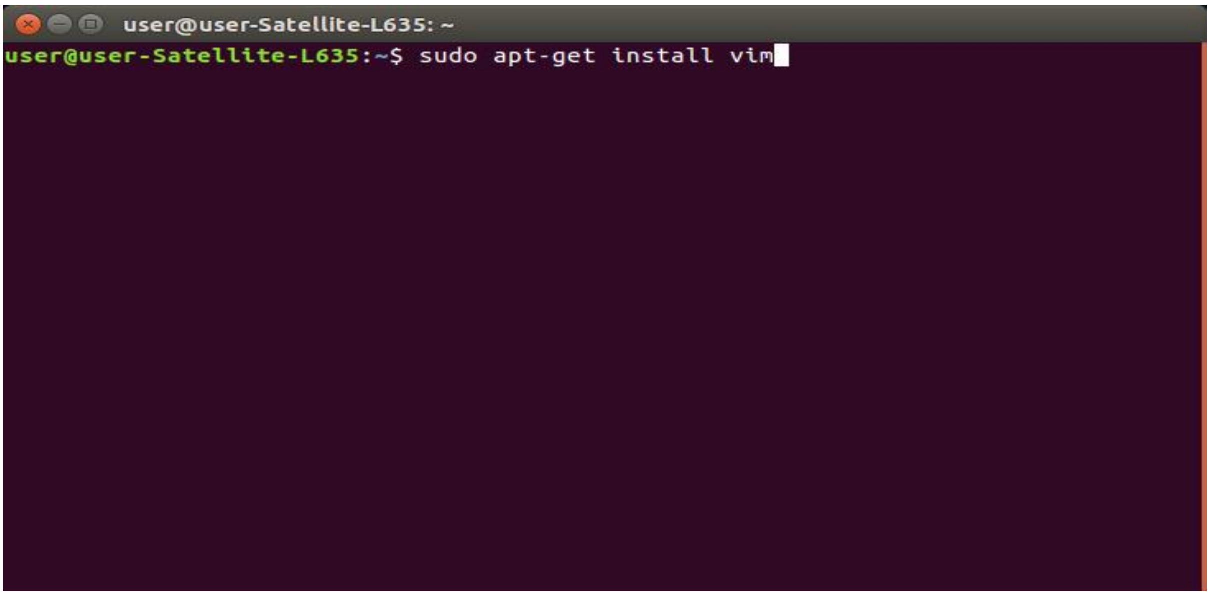


user@user-Satellite-L635: ~

```
user@user-Satellite-L635:~$ sudo apt-get update
[sudo] password for user: 
```

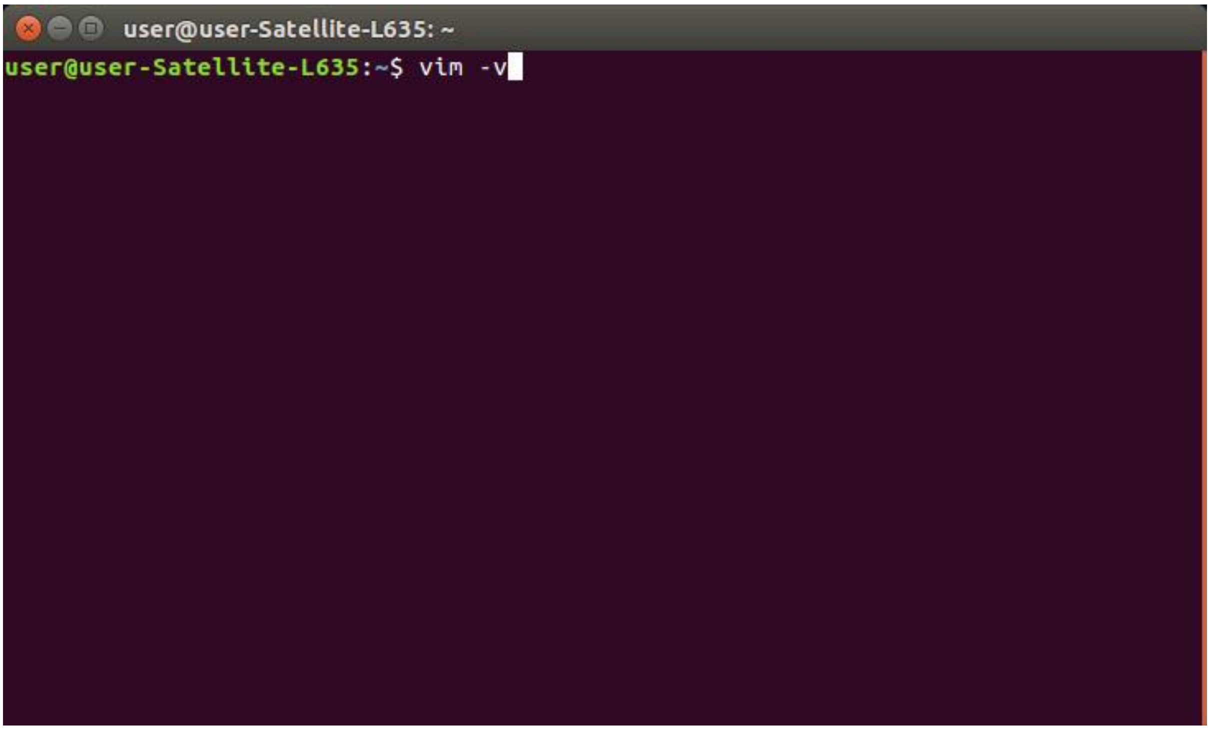
Step 2:Type the following apt-get command to install vim text editor (when promoted type your own password):

sudo apt-get install vim

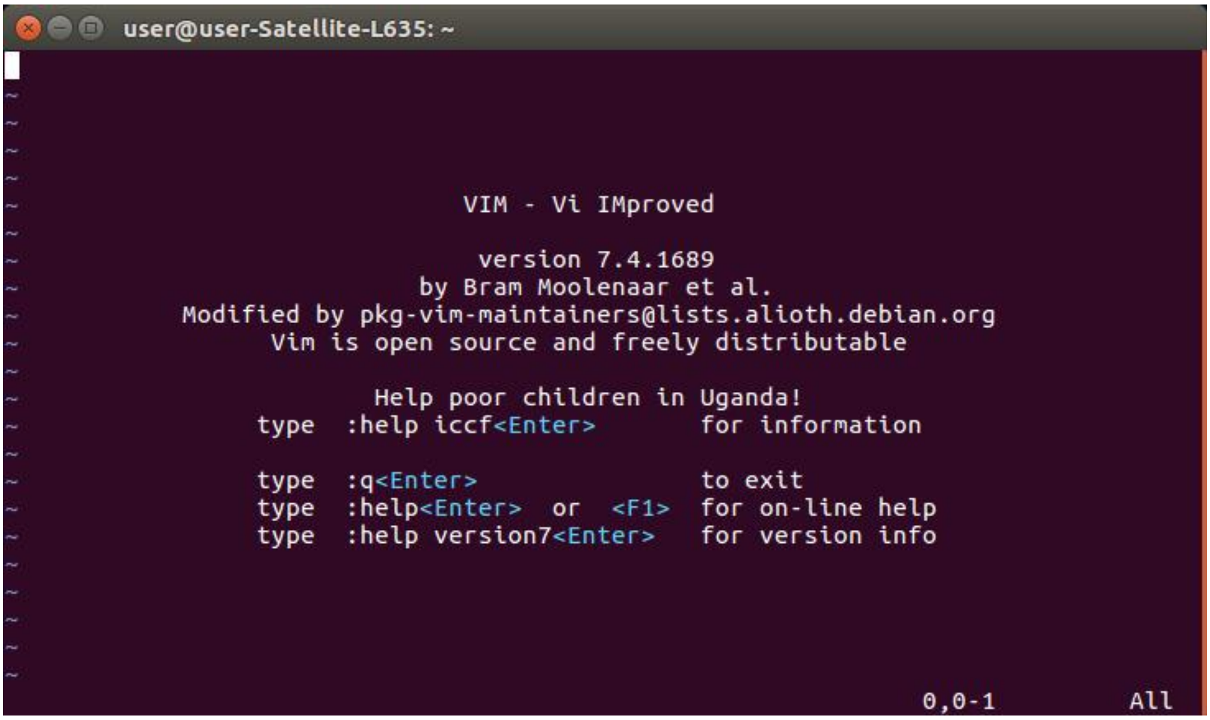


```
user@user-Satellite-L635:~$ sudo apt-get install vim
```

Step 3: Type the command ‘’vim -v’’



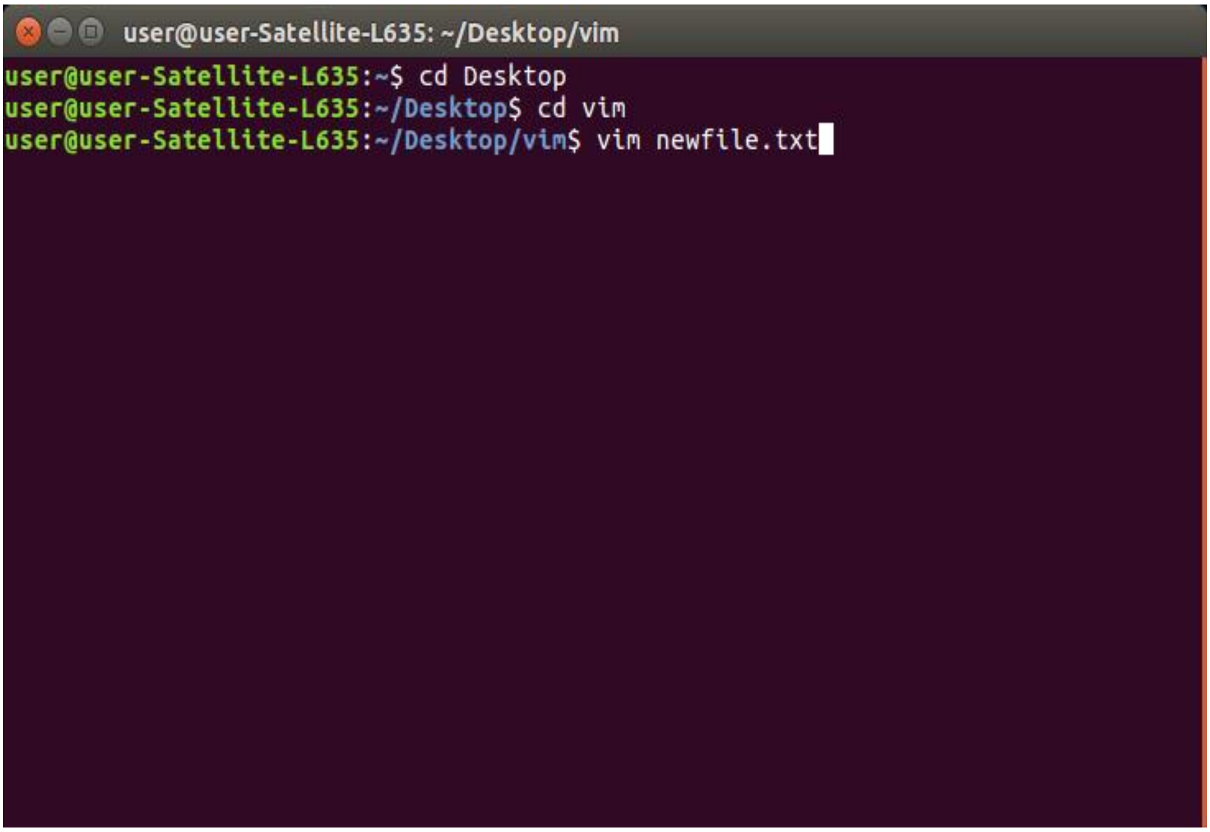
```
user@user-Satellite-L635:~$ vim -v
```



```
~
~
~
~
~                             VIM - Vi IMproved
~
~                              version 7.4.1689
~                          by Bram Moolenaar et al.
~          Modified by pkg-vim-maintainers@lists.alioth.debian.org
~                Vim is open source and freely distributable
~
~                       Help poor children in Uganda!
~              type  :help iccf<Enter>       for information
~
~              type  :q<Enter>               to exit
~              type  :help<Enter>  or  <F1>  for on-line help
~              type  :help version7<Enter>   for version info
~
~
~
~
~
                                                              0,0-1         All
```

Step 4:Create a folder named vim in your desktop.

Go to that folder using cd command.

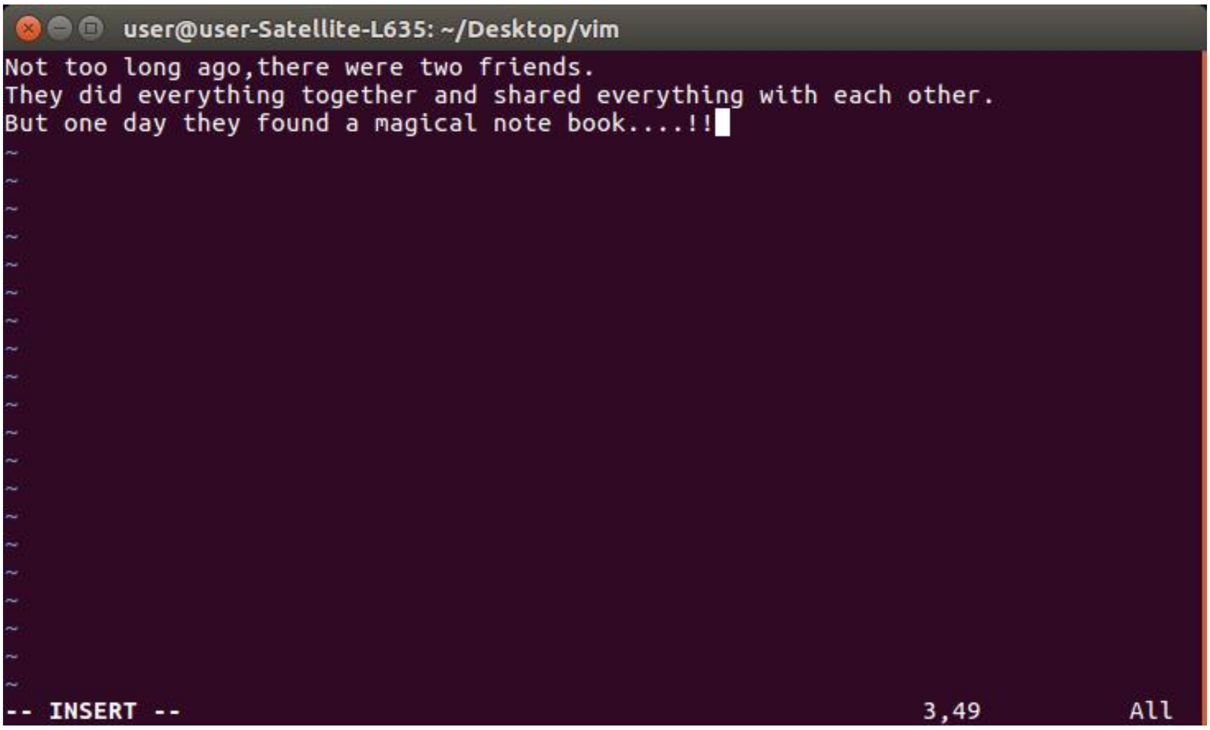


```
user@user-Satellite-L635:~$ cd Desktop
user@user-Satellite-L635:~/Desktop$ cd vim
user@user-Satellite-L635:~/Desktop/vim$ vim newfile.txt
```

Step 5: Creating a document with Vim.To create a document and start editing it, just run the vim command followed by the file name:

```
vim my_file
```

When executing the command above, the “my_file” file will be generated (the extension hasn’t been indicated but can optionally be defined, i for example, by typing: vim newfile.txt).An editor, where writing the content of the document, will immediately open.By using insert mode “i” we can insert text to the editor.



```
Not too long ago,there were two friends.
They did everything together and shared everything with each other.
But one day they found a magical note book....!!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                          3,49          All
```

Once you open the document, start writing the content. Once you have finished writing, press "Esc" and then type ": wq". Press "Enter" and your file will be saved in the path where you started the command (the meaning of ": wq" will be explained later).

This is the easiest way to use Vim as a text editor, but this tool has much more potential to show! !

VIM TEXT EDITOR MODES

The main difference between Vim and other text editors is that this tool can be considered as a 'modal' editor: depending on how you are using Vim, the same key combination can take on different functions. There are three modes:

i .Normal

ii .Insert

iii.Command-line.

Exit from any mode by pressing the ESC key, through which you will enter the "command line" mode.

i.Normal mode

With the normal mode it is possible to edit a text or navigate through the lines. In this mode, by pressing the h , j , k , l keys, you move through the text just as if you were pressing the directional arrows on the keyboard. More precisely:

h: move the cursor to the left

j: move the cursor down

k: move the cursor up

l: move the cursor to the right

Each command can be executed by placing a number next to it. For example the 3k command will move the cursor 3 lines up.

ii.Insert mode

The insert mode is the one through which Vim is used as a normal text editor, thus allowing you to add text, delete it, etc. Also, in this mode there are key combinations for inserting text in different points of the document.

To exit the entry mode, simply press the "ESC" key.

To enter this mode, just type the command: i .

iii.Command line mode

From this mode more complex commands, such as saving the changes made to the document or even closing Vim, can be executed.

These commands must be preceded by : (colon). Again, macros (combinations of commands) to be executed in series can be created.For example, if you want to save and close a document, use the command : wq where:

"W" (write) represents the request to write the changes made (save)

"Q" (quit) is used to close the document.

These commands can be combined to use the potential of this editor.

VIM COMMANDS

Basic commands

Esc - Exits current mode into the “command mode”

i - Exists current mode into the “insert mode”

:help <keyword> - Searches the Help documentation for your keyword

:w - Saves your file

:wq - Saves and closes your file

:q - Closes your file

ZZ - Saves your file and exits Vim

Navigating commands

h - Moves the cursor to the left

j - Moves the cursor down one line

k - Moves the cursor up one line

l - Moves the cursor to the right

10j - Moves the cursor down 10 lines

H - Moves cursor to the top line on the screen

M - Moves cursor to the middle line on the screen

L - Moves cursor to the bottom line on the screen

w - Moves the cursor to the beginning of the next word

b - Moves the cursor to the beginning of the previous word

e - Moves the cursor to the end of the current word

gg - Moves cursor to the first line of the file

G - Moves cursor to the last line of the file

0 - Moves cursor to the beginning of the current line

# - This command takes you to line #, where # is specified by you

Editing commands

i - Insert before the current character

a - Insert after the current character

o - Insert a line below the current line, then enter insert mode

O - Insert a line above the current line, then enter insert mode

s - Delete character at cursor and insert

S - Delete line at cursor and insert

. - Repeat last command

r - Replace one character and return to command mode

u - Undo

Searching commands

/<keyword> - Searches document for where the keyword is

/word - Finds the next instance of ‘word’

* - Finds the next instance of the current word

# - Finds the previous instance of the current word

n - Searches your text again in the direction of the last search

Working with multiple files

:bn - Moves to next buffer

:bp - Moves to previous buffer

:bd - Deletes a buffer

:sp <filename> - Opens a file in a new buffer and splits screen horizontally

:vsp <filename> - Opens a new file in a new buffer and splits the screen vertically

ctrl + ws - Split windows

ctrl + ww - Switch between windows

ctrl + wq - Quit a window

ctrl + wv - Split windows vertically

Marking Commands

v - Starts visual mode, marks lines, and runs a command

V - Starts linewise visual mode

o - Moves to the other end of marked area

ctrl + v - Starts visual block mode

aw - Marks a word

What makes Vim so popular is that it’s incredibly functional without having to use a mouse. As you can see, there are tons of keyboard shortcuts that Vim offers..