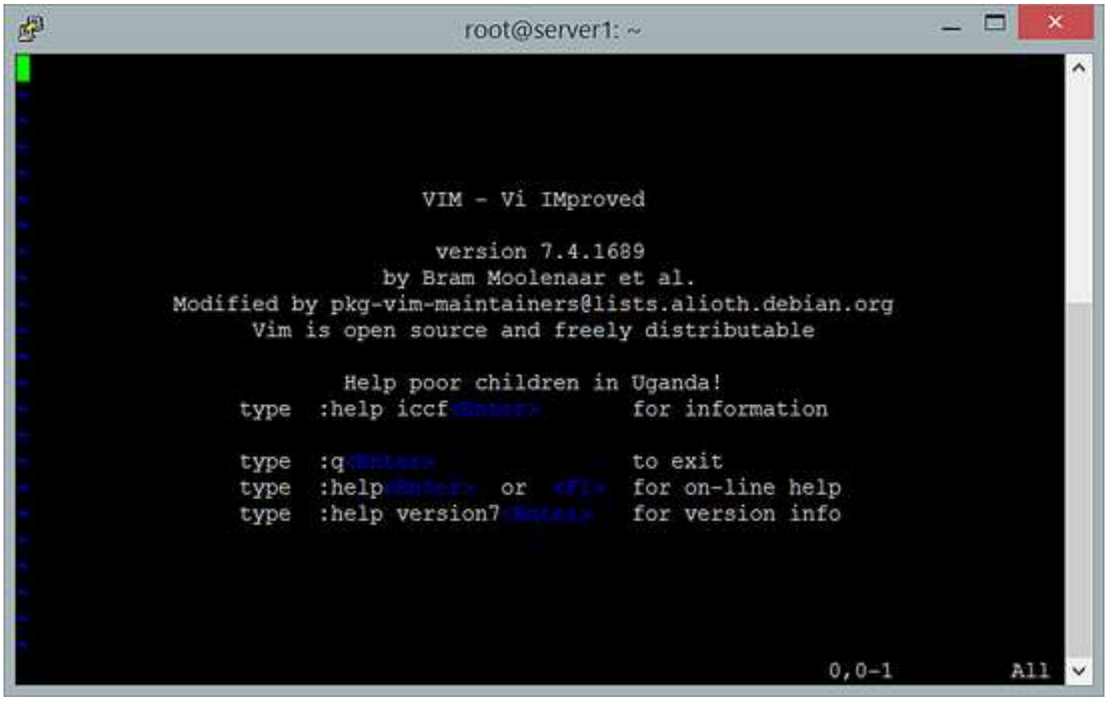
Vim Basics

This tutorial is going to speak about vim basic use. Vim is a powerful text editor used in CLI (command line interface). Linux uses a lot of configuration files, you'll often need to edit them and vim is a great tool to do so. Alternatives to vim are the commandline editors nano and joe.



```
                    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
```

Vim has a particular working method, there are two main modes: the command mode and the other modes.

The command mode lets you select the working mode that you want to enter. Available modes are: **save, quit, copy, paste** and that kind of things but you can't edit the file in the command mode directly. This is what many users that are new to vim puzzles and one has to get used to first.

Vim modes

There are several other modes, I'll cover only the most widely used ones here. At the end of the tutorial, you will find some links to in-depth guides about VIM and explanations of advanced modes.

Insert Mode

The Insert mode lets you insert text in a document. The shortcut is: "**i**" (insert text where the cursor is) or "**o**" (insert text at the beginning of the following line).

Visual Mode

The visual mode permits the user to select the text like you would do with a mouse, but using the keyboard instead of the mouse. Useful to copy several lines f text for example. The shortcut is: "**V**".

Command Mode

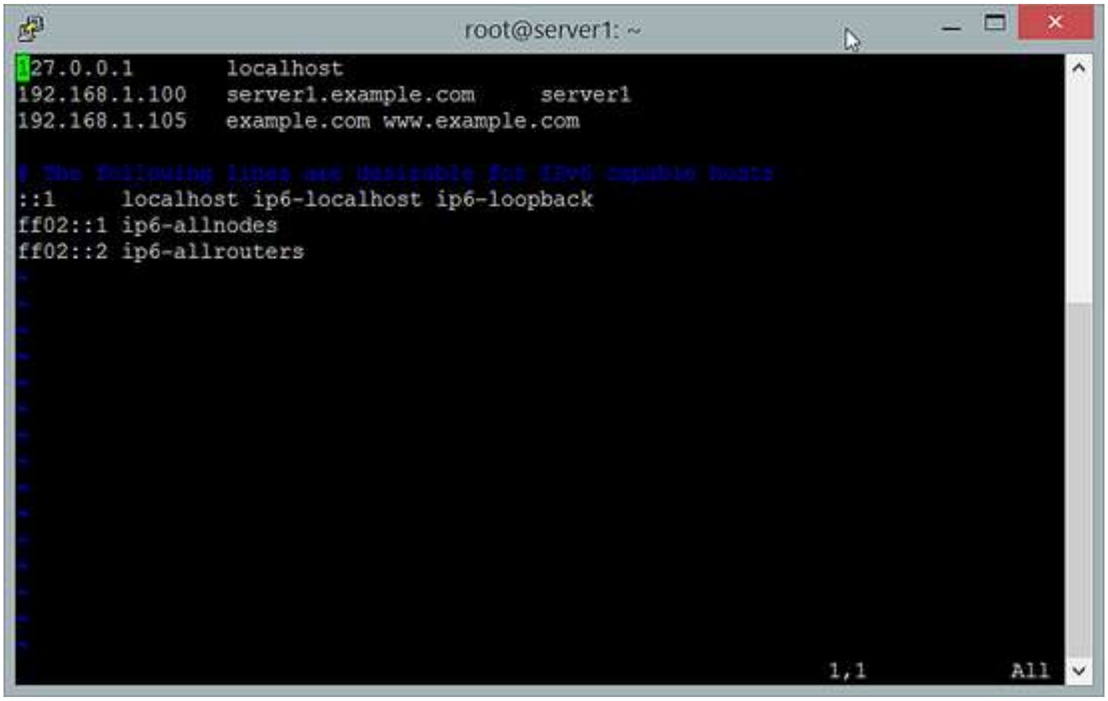
Let's now speak about the command mode, a command begins with the symbol ":".
When you are in another mod you can use the escape key (sometimes you'll need to hit it twice) to come back to command mod at any time.
For a more in-depth explanation of the Vim modes, take a look at the tutorial: Vim Editor Modes Explained

Vim usage example

To start using vim, just run the "vim" command on the Linux shell followed by the path of the file that you want to edit.
Example, editing of the file /etc/hosts

```
vim /etc/hosts
```

The result will look like this:



```
127.0.0.1       localhost
192.168.1.100   server1.example.com     server1
192.168.1.105   example.com www.example.com

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                              1,1           All
```

The editor is now in command mode. To start editing the file content, enter:

```
:i[enter]
```

[enter] means to press the return or enter key on your keyboard.
The word --insert-- will appeer at the bottom of the editor window to show that you are in insert mode now.

```
-- INSERT --
```

Now you can edit the file by navigating to the line that you want to change with the cursor keys and then start typing the text. When you are finsihed with editing, press the [esc] key to go back to the command mode.

To save the file and exit the editor, enter:

```
:x[return]
```

In case you want to quit vim without saving the file, enter:

```
:q![return]
```

Vim Command Reference

save: **:w**
save and exit: **:wq**
exit: **:q**
force: **!** (example **:w! :q!**)
vertical split: open a document and then type :vsplit /path-to-document/document and this will open the specified document and split the screen so you can see both documents.
copy: **y**
copy a line: **yy**
paste: **p**
cut: **d**
cut a line: **dd**