Linux Beginner Guide

Jaewoong Lee

Ulsan National Institute of Science and Technology

jwlee230@unist.ac.kr

January 5, 2020

Introduction

In this guide, I assume that followings are already installed:

1. Ubuntu 16.04.2 or Higher
2. ZSH 5.0.2 or Higher
3. VIM 8.1 or Higher
4. We will connect to server via SSH

With this guide, you can use and understand Linux system.
Also, this guide includes as little information about operating system as possible. If you find some fault in the strict sense of the word, that means you are not **beginner**.

Overview

Linux?

Figure: Linus Torvalds, Inventor of Linux

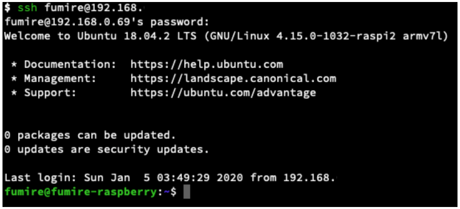
Linux is one of the most famous OS as Windows and macOS.
Linux is open-source project.
Android, OS for mobile, is based on Linux.

Ubuntu?

Figure: Logo of Ubuntu

Ubuntu is an OS which is based on Linux.
Ubuntu is the best OS in Linux-like OS, because of convenience of its installation and usage.

Where we start

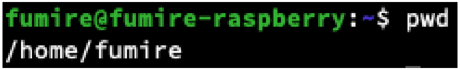


```
$ ssh fumire@192.168.
fumire@192.168.0.69's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-raspi2 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


0 packages can be updated.
0 updates are security updates.

Last login: Sun Jan  5 03:49:29 2020 from 192.168.
fumire@fumire-raspberry:~$ 
```

Figure: Here is where we start

After you connect to server via SSH, you can see like this.
Here is where we start!
fumire will be user name, and *fumire-raspberry* will be server name.

pwd

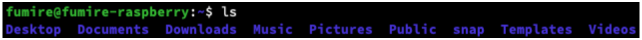


```
fumire@fumire-raspberry:~$ pwd
/home/fumire
```

Figure: Result of *pwd* Command

pwd is abbr. of "Print Working Directory".
You can see where you are with *pwd* command.
Also, "/home/username" is your *home folder*, a.k.a. '∼'.

ls



```
fumire@fumire-raspberry:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
```

Figure: Result of *ls* Command

ls stands for "List".
ls command lists current directory contents.
If current directory is empty, the result will be nothing.

Configuration

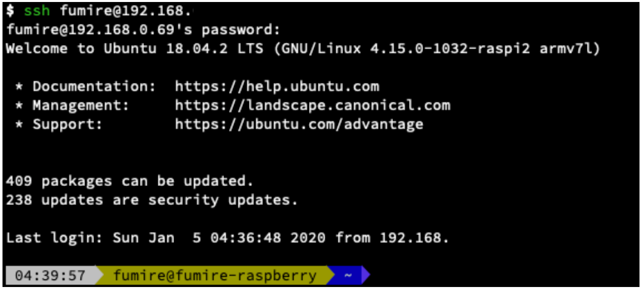
However, you have not completed configuration. Therefore, finish settings with following command:

Example

```
$ git clone https://github.com/Fumire/.dotfiles.git
$ cd .dotfiles
$ make
$ chsh -s /usr/bin/zsh
```

Note that you should input command only after '$'.
After executing commands, you should restart your shell.

Configuration *(Cont.)*

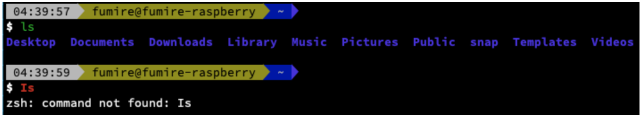


```
$ ssh fumire@192.168.
fumire@192.168.0.69's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-raspi2 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


409 packages can be updated.
238 updates are security updates.

Last login: Sun Jan  5 04:36:48 2020 from 192.168.

 04:39:57  fumire@fumire-raspberry  ~
```

Figure: ZSH

With successful configuration, you can see like this.

Tip!



```
04:39:57  fumire@fumire-raspberry  ~
$ ls
Desktop  Documents  Downloads  Library  Music  Pictures  Public  snap  Templates  Videos

04:39:59  fumire@fumire-raspberry  ~
$ Is
zsh: command not found: Is
```

Figure: Right Command vs. Wrong Command

You can easily know this command is right with ZSH as figure.

mkdir

mkdir stands for "Make Directory".
You can make a directory which named 'test' as following:

Example

```
$ mkdir test
or
$ md test
```

mkdir returns nothing. Literally, *mkdir* command only make directory.
You can check that the directory has been made with *ls* command.

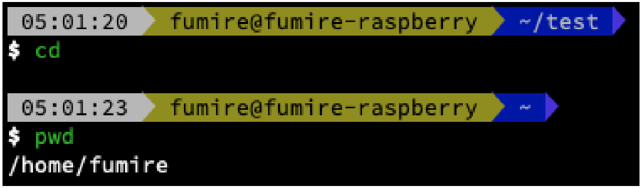
cd

cd is abbr. of "Change Directory".
You can change your working directory to 'test' as following:

Example

```
$ pwd
$ cd test
$ pwd
```

Also, you can go your home folder at once with *cd*, no matter where you are.

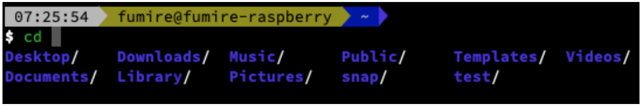


```
05:01:20  fumire@fumire-raspberry  ~/test
$ cd

05:01:23  fumire@fumire-raspberry  ~
$ pwd
/home/fumire
```

Figure: *cd* will guide you to home folder

Tip!

If you hit "Tab" button, ZSH will give proper candidates.
Following example shows what ZSH gives.



```
 07:25:54  fumire@fumire-raspberry  ~ 
$ cd
Desktop/    Downloads/  Music/      Public/     Templates/  Videos/
Documents/  Library/    Pictures/   snap/       test/
```

Figure: Shortcut with Tab

man and --help

You can get detailed information about command as following:

Example

```
$ man ls
and/or
$ ls --help
```

This guide will give simple information about Linux command. Hence, when you have curiosity about command, use these command.

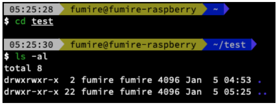
Directory Structure

Try following commands:

Example

$ cd test
$ ls -al

Then, you can see like this:



```
05:25:28  fumire@fumire-raspberry  ~
$ cd test

05:25:30  fumire@fumire-raspberry  ~/test
$ ls -al
total 8
drwxrwxr-x  2 fumire fumire 4096 Jan  5 04:53 .
drwxr-xr-x 22 fumire fumire 4096 Jan  5 05:25 ..
```

Figure: Result of *ls* command

All directory has '.' and '..', even though the directory is empty.
'.' means current directory itself; and, '..' means parent directory.

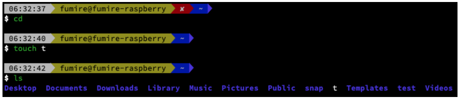
touch

touch command make new file or touch the file.
Try following example:

Example

```
$ cd
$ touch t
$ ls
```

Then, you can see that the file which name 't' has been made.



```
06:32:37  fumire@fumire-raspberry  ✘  ~
$ cd

06:32:40  fumire@fumire-raspberry  ~
$ touch t

06:32:42  fumire@fumire-raspberry  ~
$ ls
Desktop  Documents  Downloads  Library  Music  Pictures  Public  snap  t  Templates  test  Videos
```

Figure: Result of *touch* Command

mv

mv command moves/renames file. *mv* is used as:

Example

```
$ mv SRC(source) DST(destination)
```

Try following commands:

Example

```
$ mv t tmp
$ ls
$ mv tmp test/
$ ls
```

Then, you will realize that the file 'tmp' is gone. I hope that you already know where the file goes. :)

cp

cp command copies SRC to DST. *cp* is used as:

Example

```
$ cp SRC DST
```

Try following commands:

Example

```
$ cd ~/test/
$ ls
$ cp tmp tmp2
$ ls
```

Then, you can realize that a new file 'tmp2' has been made.

rm

rm stands for 'Remove'. As its name, you can delete files or directory.

Example

```
$ rm tmp2
```

When you want delete directory, use '-r' option:

Example

```
$ rm -r directoryname
```

There is no way to restore removed files!! Beware what you remove!!

sudo

sudo is abbr. of "Substitute User do"; but, many people know as "Super User do".
sudo allows a system administrator to delegate authority to give certain user the ability to run some command as another user.

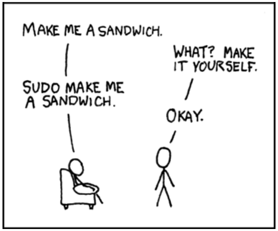


Figure: XKCD: Sandwich

THINK what will happen after *sudo* command!!

Editor

There are three major editors in Linux.

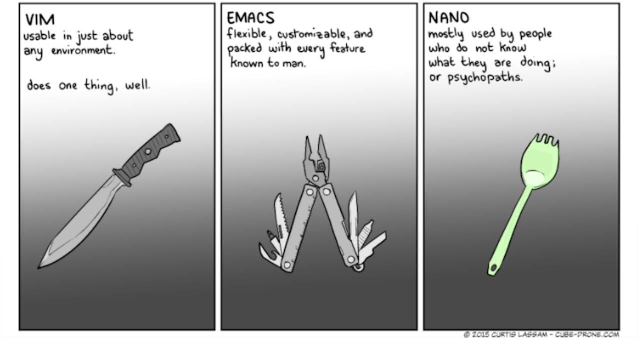

VIM
usable in just about any environment.
does one thing, well.
EMACS
flexible, customizable, and packed with every feature known to man.
NANO
mostly used by people who do not know what they are doing; or psychopaths.
© 2015 CURTIS LASSAM - CUBE-DRONE.COM

Figure: Descriptions of Editor

For this reason, this guide use VIM editor.

Editor *Cont.*

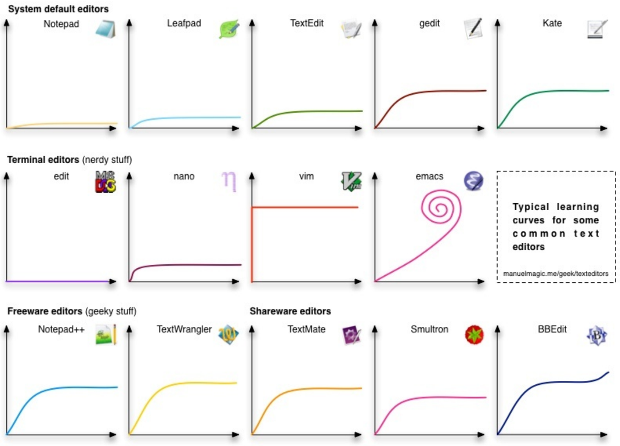


Figure: Learning Curves among Editors

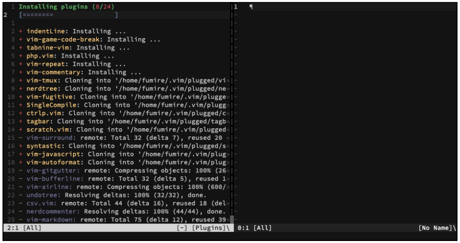
First Meet with VIM

With these commands, you can make/edit file.

Example

```
$ vi tmp
```

If it is first time to open VIM, then you will see like this.



```
Installing plugins (8/24)
[========                 ]

+ indentLine: Installing ...
+ vim-game-code-break: Installing ...
+ tabnine-vim: Installing ...
+ php.vim: Installing ...
+ vim-repeat: Installing ...
+ vim-commentary: Installing ...
+ vim-tmux: Cloning into '/home/fumire/.vim/plugged/vi»
+ nerdtree: Cloning into '/home/fumire/.vim/plugged/ne»
+ vim-fugitive: Cloning into '/home/fumire/.vim/plugge»
+ SingleCompile: Cloning into '/home/fumire/.vim/plugg»
+ ctrlp.vim: Cloning into '/home/fumire/.vim/plugged/c»
+ tagbar: Cloning into '/home/fumire/.vim/plugged/tagb»
+ scratch.vim: Cloning into '/home/fumire/.vim/plugged»
- vim-surround: remote: Total 32 (delta 7), reused 20 »
+ syntastic: Cloning into '/home/fumire/.vim/plugged/s»
+ vim-javascript: Cloning into '/home/fumire/.vim/plug»
+ vim-autoformat: Cloning into '/home/fumire/.vim/plug»
- vim-gitgutter: remote: Compressing objects: 100% (26»
- vim-bufferline: remote: Total 32 (delta 5), reused 1»
- vim-airline: remote: Compressing objects: 100% (600/»
- undotree: Resolving deltas: 100% (32/32), done.
- csv.vim: remote: Total 44 (delta 16), reused 18 (del»
- nerdcommenter: Resolving deltas: 100% (44/44), done.
- vim-markdown: remote: Total 75 (delta 12), reused 39»
2:1 [All]                        [-] [Plugins]\ 0:1 [All]                        [No Name]\
```

Figure: First Time of VIM

Modes of VIM

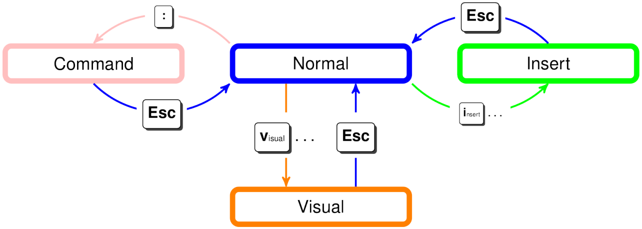


Figure: Three Modes in VIM

How to Edit with VIM

Editing with VIM is following such steps:

1. Press 'i'
2. Edit the file
3. Press 'ESC'
4. Enter ':w' which means *write*
5. Enter ':q' wihch means *quit*

Plugin Setting (Optional)

You might see the error message because the plugin setting is not completed. To solve this, use following commands:

Example

```
$ cd ~/.vim/plugged/tabnine-vim/
$ python3 install.py
```

Then, all plugin acts properly without errors.

cat

cat stands for con**cat**enate. *cat* command reads files, and writing them to standard output.
Consider following example:

Example

```
$ cd ~/test/
$ cat tmp
```

You can see contents of file.

Input/Output to file

If you want redirect output to file, use following example:

Example

```
$ cat tmp > output
```

However, this method *overwrites* the contents of file.
If you want preserve the file contents, use following:

Example

```
$ cat tmp >> output
```

Also, < means input from file.

Output to file *Cont.*

In some cases, you should divide standard output and standard error.
In these cases, use following commands:

Example

```
$ commands 1> STDOUT 2> STDERR
```

Example

```
$ cat tmp 1> ex.stdout 2> ex.stderr
```

more / less

more and *less* are commands for seeing the contents of file. Consider following examples:

Example

```
$ more tmp
```

Example

```
$ less tmp
```

Pipe

Use pipe (|) to indicate input as output of previous command.

Example

```
$ command1 | command2
```

The output of command 1 will be the input of command2.
Consider following example:

Example

```
$ cat tmp | less
```

Two Ways for Download

There are two main ways for download.

1. curl
2. wget

curl

Example

```
$ curl https://www.naver.com
```

curl returns to standard output. If you want to get file, consider following:

Example

```
$ curl https://www.naver.com -o naver.html
```

wget

wget returns a downloaded file as output.

Example

```
$ wget https://www.naver.com
$ ls
```

gzip

gzip is used for file compression and decompression.
When compressing:

Example

```
$ gzip tmp
```

When decompressing:

Example

```
$ gzip -d tmp.gz
```

However, the examples hereinabove delete the original files. If you want *keep* original file, consider following:

Example

```
$ gzip -k tmp
```

TAR Files

TAR stands for "Tape Archives".

Originally, it is used for tape; but, in nowadays, it is used for file archiving system. Usually, make a directory to one TAR files.

TAR.GZ file is commonly used for distribution some software. For example:

Example

```
$ wget https://www.python.org/ftp/python/3.8.1/Python-3.8.1.tgz
```

(TGZ is for TAR.GZ)

TAR Files *(Cont.)*

You might think decompress TGZ file and make a directory from TAR file. However, you can decompress TGZ file at once:

Example

```
$ tar -zxvf Python-3.8.1.tgz
```

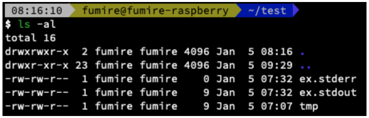
Then, you can see the directory named 'Python-3.8.1".

Permissions

With the following command, we can know how permissions are set:

Example

$ ls -al



```
08:16:10  fumire@fumire-raspberry  ~/test
$ ls -al
total 16
drwxrwxr-x  2 fumire fumire 4096 Jan  5 08:16 .
drwxr-xr-x 23 fumire fumire 4096 Jan  5 09:29 ..
-rw-rw-r--  1 fumire fumire    0 Jan  5 07:32 ex.stderr
-rw-rw-r--  1 fumire fumire    9 Jan  5 07:32 ex.stdout
-rw-rw-r--  1 fumire fumire    9 Jan  5 07:07 tmp
```

Figure: File Permissions

The way to read this result is following:

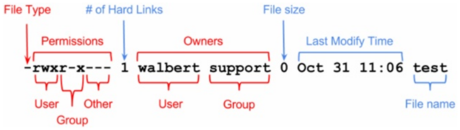


Figure: How to Read Permissions

chmod

chmod stands for "Change Mode". You can modify permissions of files. Before input command, you should calculate simple arithmetic:

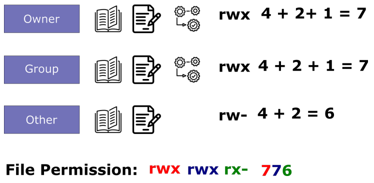


Figure: Simple Arithmetic

Then, you will get three digits for permission. Moreover, 660 or 770 are usually used.

Example

```
$ chmod three_digits filename
```

chmod *(Cont.)*

Or, you can do as followings:

Example

```
$ chmod rwx------ tmp
```

Example

```
$ chmod g=rwx tmp
```

Example

```
$ chmod +x tmp
```

chown

chown stands for "change ownership". As its name, you can modify ownership of file.

When you want to change only USER:

Example

```
$ chown USER tmp
```

When you want to change both USER and GROUP:

Example

```
$ chown USER:GROUP tmp
```

Ctrl-C

When you have started command, but you realize that you should stop the command, then use "Ctrl-C".

Example

```
$ sleep 99999
$ ^C
```

Ctrl-C sends SIGINT, which stands for "Signal Interruption"; and, the process is going to terminate after receiving SIGINT.

&

Consider long-time procedure, such as:

Example

```
$ sleep 99999
```

However, with '&', you do not have to wait procedure.

Example

```
$ sleep 99999 &
```

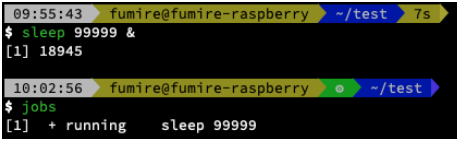
Then, the process is executing on background.

jobs

jobs commands shows the background process.

Example

$ jobs



```
09:55:43  fumire@fumire-raspberry  ~/test  7s
$ sleep 99999 &
[1] 18945

10:02:56  fumire@fumire-raspberry  ⚙  ~/test
$ jobs
[1]  + running    sleep 99999
```

Figure: Result of *jobs*

You use record the [number] for handle the process.

kill

kill command kills the process.
When you want to kill background process, consider following:

Example

```
$ kill %number
```

The number is from the *jobs* command.

Example

```
$ kill process_number
```

You can kill as above when you know exact process number (PID).

nohup

When you lost from SSH connection, all executing process receive SIGHUP, which stands for Signal Hangup, and will be terminate even they runs on background.
To prevent SIGHUP, use *nohup* command.

Example

```
$ nohup sleep 99999 &
```

However, *nohup* command makes 'nohup.out' automatically. You can change output file name with IO redirection.

screen

screen prevents unintentional connection lost.
Make screen session with simple command:

Example

```
$ screen
```

When you want to detach the screen session, use following, instead of *exit*:

Example

```
$ ^a, d
```

When you restore the screen session, use following command:

Example

```
$ screen -r
```