Ubuntu - Commands in detail

Usman Wajid
usman.wajid88@gmail.com

1 Commands

A command is a request from a programmer, an operator, or a user to Linux operating systems asking that a specific function be performed. For Example, a request to list all files in your current directory will be the command ls.

Syntax

The general way commands are entered in Linux is as such:

```
command -option(s) argument(s)
```

Here,

- **command:** A command tells the operating system what to do (what action to be performed, copy a file, display a date etc.)
- **option(s):** It tells the way of action to be per- formed. For example, ls command displays directory contents, and -r option tells the way in which the directory should be displayed. Here **-r** displays directory contents in reverse (alphabetically) order.
- **argument(s):** Argument tells that on what objects (file, directory, devices, etc.) the command and its arguments are applied. For example if we need to display all files starting with alphabet a, you will give "**ls a***" and press enter.

1.1 The Asterik *

The asterisk * symbol is basically a wildcard. It can be used in a number of contexts. For example:

- It can be used to denote *everything*. For example, in MS-DOS, typing *delete* * will delete all files in a current directory. With Linux, you can use *rm* * to do the same thing.
- It can be used as a filter. For example, typing *ls* ab* will print all file/folder names that start with ab.

1.2 Case Sensitivity

Linux Commands are case-sensitive. All standard Linux commands are given in lower case letters only. As an example, typing ls will print the directory contents. Typing Ls, or LS, or lS will result in a command syntax error.

1.3 Auto-Completion

Auto-Completion is a short-cut feature for quickly entering commands that are long or you have forgotten their spelling. To practice, just type f and press the TAB key. You will see the list of all commands starting with an f. Type fd and press TAB, you will see all commands staring with fd. Type fdi and press TAB, you will see a list of commands all starting with fdi, so on and so forth.

You can also use the auto-completion to detect directories. For example, you want to access the home directory of a user who for some strange reason is named abcdefghijklmnopqrstuvwxyz. From the root directory (/), you will type cd /home/a and press TAB. The rest of the characters bcdefghijklmnopqrstuvwxyz will be given automatically and you will be spared the time and effort of writing such a large name.

1.4 Redirection

You can use the > and < symbols to redirect your output. The types of redirection are as such:

- > Output redirection to a file
- 1> Same as >
- 2> Error output redirection to a file
- < Output redirection from file to a terminal

Try it using the following set of Commands

```
cd

ls

touch newfile

ls

ls > newfile

cat < newfile

ls -lh

ls -lh 1> newfile

lsot

lsot 2> newfile

cat < newfile

rm newfile
```

1.5 Practicing Commands

Some of the most commonly used commands are given below. Try and practice each one of them and see what they do.

ls Print the contents of the current directory (or any other directory is path is specified)

cd Change directory

mkdir Create new directory

rmdir Delete an empty directory

cat View contents of a file, or write contents to a file

cp Copy a file from one location to another

mv Move a file from one location to another

rm Remove file(s) and/or Directory(ies)

To see them working, practice the following set of commands, the # (root or superuser mode) sign represents the shell prompt.

```
$mkdir temporary

$cd temporary

temporary$ls

temporary$cat > newfile

Type any text and press Enter and then press CTRL+D

temporary$cat newfile

temporary$mkdir another

temporary$ls

temporary$cp newfile newest

temporary$ls

temporary$cp newfile newester

temporary$ls

temporary$cd another
```

```
another$ls

another$cp newest newestest

another$cat newestest

another$cd ..

temporary$mv newester another/newester

temporary$ls

temporary$ls another/n*

temporary$cd ..

$rm temporary

$rm temporary/*

$rm -rf temporary
```

2 Absolute vs Relative path

while using commands like cd, mkdir, or mv etc we can provide either absolute or relative paths

2.1 Absolute path

It always start with the / symbol. It means that we have to provide the complete path from the root directory '/' upto the last directory. The following commands shows the example of absolute path
`usman@usman−HP: $ cd /home/usman/Downloads`

2.2 Relative Path

It relates path to some existing path or directory.

2.2.1 ~ symbol

The tilde symbol ~ represents the home directory that is /home/usman. We can go to our home directory by simply typing the following command

```
usman@usman−HP:/media$ cd ~
usman@usman−HP:~$ pwd
/home/usman
```

we can also move to a certain directory or directories in relation to the home directory. For example,

```
usman@usman−HP:/media$ cd ~/Downloads
usman@usman−HP:~/Downloads$ pwd
/home/usman/Downloads
```

2.2.2 Working within directory

If we are already inside a directory we only need to specify the name(s) of its subdirectory or subdirectories. As an example,

```
usman@usman-HP:~$ cd Downloads/test
usman@usman−HP:~/Downloads/test$ pwd
/home/usman/Downloads/test
```

3 Multiple arguments with a single command

We can used multiple arguments for a particular command by enclosing the multiple arguments with in curly braces and separated by commas. As an example,

```
usman@usman-HP:~/temp$ mkdir {test,test1,test2}
usman@usman-HP:~/temp$ ls
test test1 test2
usman@usman-HP:~/temp$
```

We can create multiple subdirectories within a directory that doesn't exist yet by using a -p option with the mkdir command. As an example,

```
usman@usman-HP:~/temp$ mkdir -p temp/{test,test1,test2}
usman@usman-HP:~/temp$ ls temp/
test test1 test2
usman@usman-HP:~/temp$
```