Basics on linux command line

Daniel Tiezzi

WELLCOME!

In this tutorial we are demonstrating how to work with basic shell commands.

The fist step is to understand the linux directory structure.

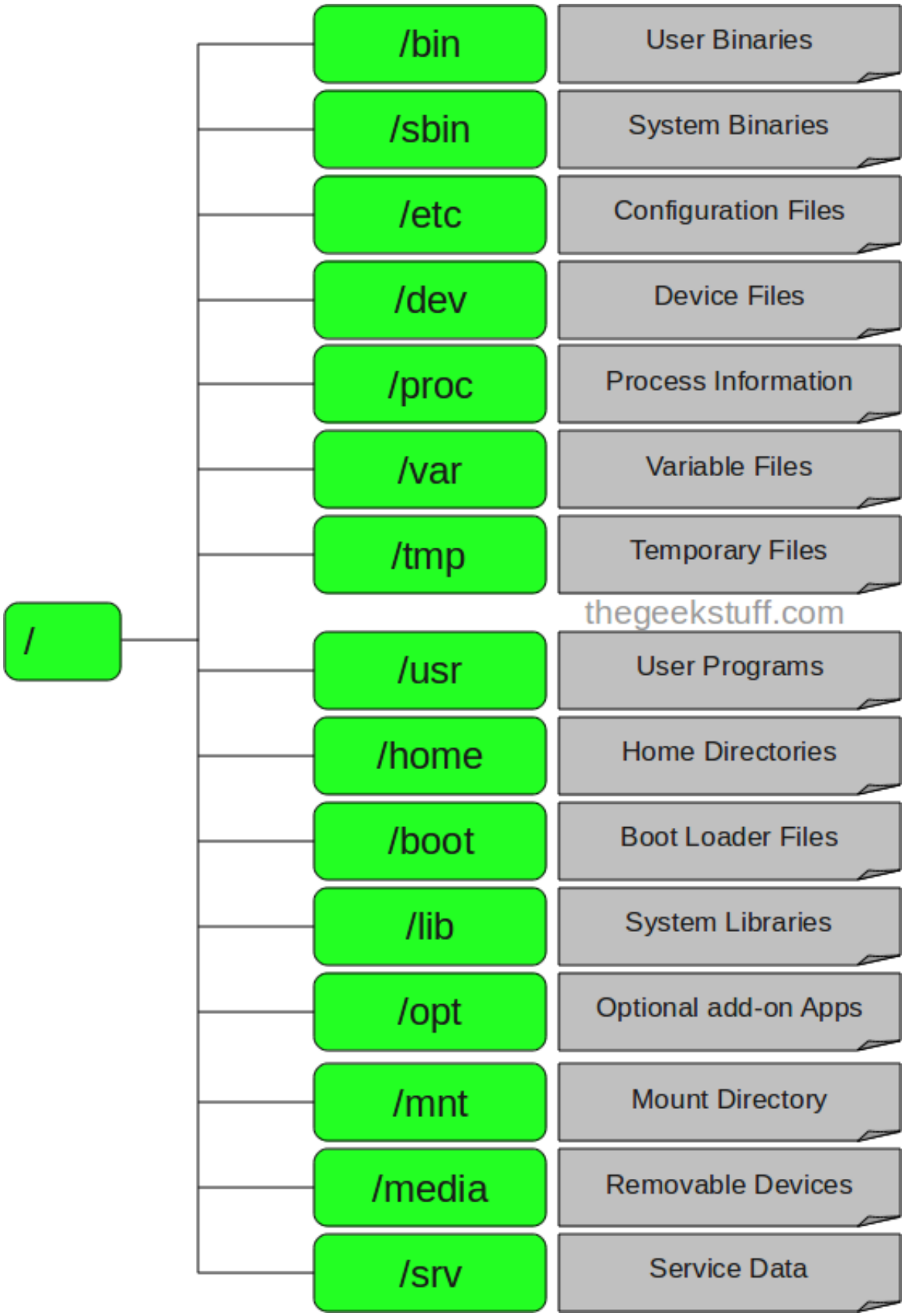
Before we start, we need to access a Terminal, the interface we are going to communicate with the computer *via* the shell command line. Shell is the interpreter for linux. That means shell is able to get the programming language you type on the keyboard and convert to computer language the operating system can handle. So, open a Terminal window. You can do it just pressing together the `Crt` + `Alt` + `t` keys.

What you see on the Terminal window is a black screen with a writing like this:

```
dtiezzi@kushin:~$
```

What does that mean? The `dtiezzi` is the username who is logged in `@` at the `kushin` machine. Then the `:~` is the directory you are working. The `~` represents the user home directory. The `$` is the prompt signal. The `$` means the shell is ready to receive your command.

The directory structure begins at the root `/` directory. There is a branching tree from the root directory. The figure bellow illustrates the linux directory structure and each directory meanings.



Note there is the `/home` directory from the root. The `/home` directory contains the home directory for all users. So, if you type `pwd` (print work directory) you will see something like this:

```
dtiezzi@kushin:~$ pwd
/home/dtiezzi
```

So, note you have jus typed your first command on shell. The interpreted understood your `pwd` command and returned on the screen your actual work directory.

The other important command is `ls` (*list* files). So, typing `ls` you see something like this:

```
dtiezzi@kushin:~$ls
Desktop    Downloads  Pictures  R         Templates
Documents  Music      Public    Software  Videos
```

The command to change directories is `cd`. So, to change to `Documents` just:

```
dtiezzi@kushin:~$ cd Documents
dtiezzi@kushin:~/Documents$ pwd
/home/dtiezzi/Documents
```

To update softwares and install new ones you can use the command line as well. The command `apt-get` does this. So, first type `sudo apt-get update` to update your already installed packages. The `sudo` is a command to run as a administrator. So you will be asked to type the admin password. You have to type and press `enter`. Note when you type the password in linux nothing is displayed on the screen. No worries, just type it and `enter`.

After updating, you will install a new software. Let's install the `tree` software. The tree software allows you to visualize the tree directory in your computer. So, just type `sudo apt-get install tree`. After installed, type `tree -L 1 /home/USERNAME/` and you will see something like this:

```
tree -L 1 /home/dtiezzi/
/home/dtiezzi/
├── Desktop
├── Documents
├── Downloads
├── Music
├── Pictures
├── Public
├── R
├── Software
├── Templates
└── Videos
```

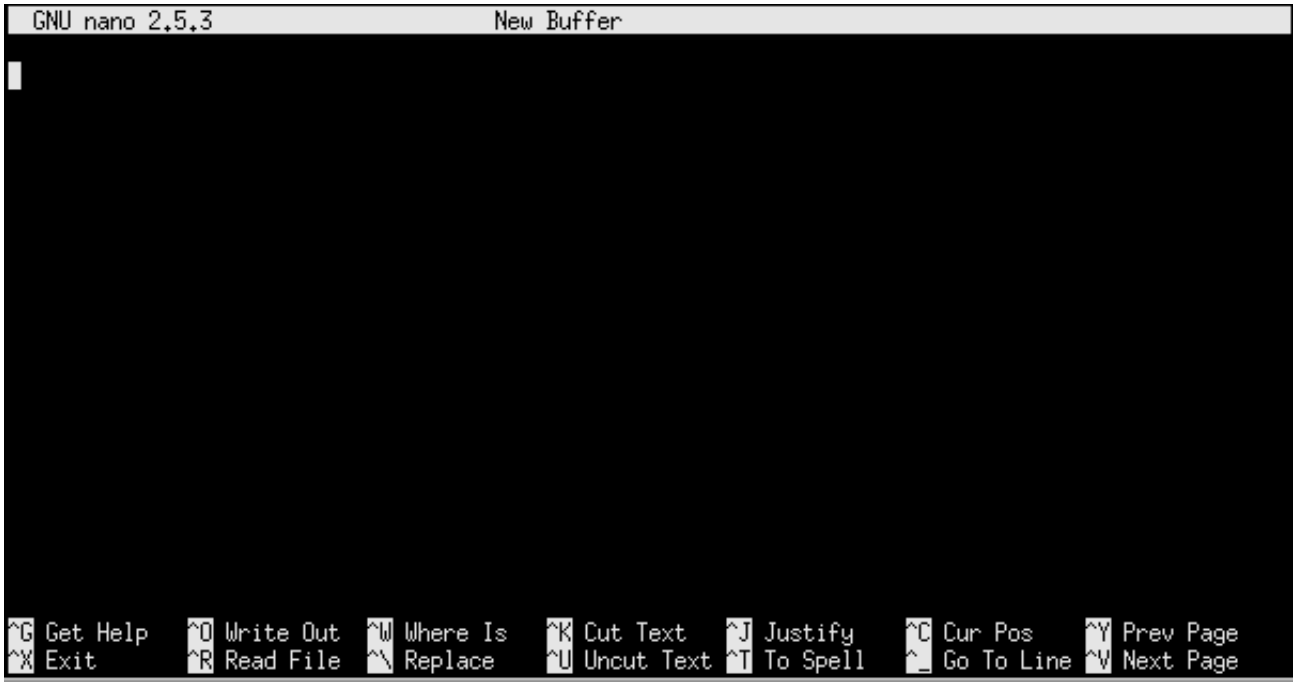
`-L 1` is a option for the `tree` command. It makes `tree` displays only one level ahead of the

`/home/dtiezzi` directory. So, if you use `-L 2`, you have a new branching level and so on. If omit, the complete directory branch will be displayed.

Now are writing your first software in linux. The executable file in linux have a `.sh` extension. To write a software, you need a text editor. We will be using the `nano` text editor. You can access `nano` on the Terminal.

First, we will be creating a new directory. So, if you are in the `Documents` directory, just type `mkdir shellScripts` (mkdir = make directory) and press `enter`. Now use `cd shellScripts` to go to the new directory where we are going to write the software.

Now type `nano greet.sh` and you will see something like this:



This is the `nano` editor where you can type anything you want. However, we are writing the software. Do, just copy the piece of code bellow on your `nano` editor:

```bash
#!/bin/bash

read -p "Enter your name" name
echo "Hi $name. Let us be friends!"
```

Let's understand this piece of code.

`#!` is called shebang. When it is the first two bites of an executable file, it is interpreted by the execve system, which execute the program according to interpreter executable defined by the interpreter path, here the /bin/bash. So, this makes a shell executable file.

`read -p "Enter your name" name`. `read -p` receives input from the prompt. So, whatever you

type is passed to the variable `name`. `"Enter your name"` is what is displayed, a instruction to the user to type the input data on the keyboard.

`echo "Hi $name. Let us be friends!"`. `echo` is a function that echoes what comes next, int his case `"Hi $name. Let us be friends!"`. And the `$` sign before the variable `name`? The `$` is used before a variable to access its value, in this case, the name the user has typed.

Now you need to save this file. In nano, press `Ctl` + `x` and then press `y` followed by `enter`. Now, if you `ls` you see a file `greet.sh` on your directory.

Now, type `ls -l`. The `-l` makes the `ls` output as a list. You will se a line like this:

```
-rw-r--r-- 1 root root 14 Feb 19 01:57 greet.sh
```

The `-rw-r--r--` is file's permissions. We are getting on this later, but the `r` states for read and `w` for write. So, there is no permission to `x` execute this file. You need to create a `x` permission. The `chmod` command is responsible to define files and directories permissions. So, type `sudo chmod +x greet.sh` followed by `ls -l`. Now it is like this:

```
rwxr-xr-x 1 root root 84 Feb 19 00:17 greet.sh
```

Now to execute, you just type `./greet.sh` and you have this:

```
dtiezzi@kushin:~/Documents/shellScripts$ ./greet.sh
Enter your name: Daniel
Hi, Daniel. Let us be friends!
```

To finish this tutorial, let's see how to copy `cp`, move `mv` and remove `rm` files. To copy a file from a directory to another just type:

```
cp filename /path/to/dir
```

To move:

```
mv filename /path/to/dir
```

To remove:

```
rm filename
```

DO NOT FORGET THIS: the `rm` command completely removes the file from your disk. So, if you run this command, the file will be lost forever. So, beware.

In this tutorial we described the linux directory structure and demonstrated some basic shell commands to work on terminal, an interface between the user and the computer. Now you have learnt how to navigate over the directories, create a new one, to list the directory content, copy, move and remove files, and install

softwares on linux command line. Most exciting was to write the first software on linux and to execute it.

Now, we have some exercises you can do to fix what you have learnt and the video on the next section will cover some additional functions to work with text files.

CONGRATS!