

# LINUX COMMANDS

Daniel Felipe Vargas Arias

# Contents

1	Environment variables	2
2	Commands	3
3	Git	5
4	Java	6
5	Secure Shell (SSH)	7
6	Keyboard shortcuts	8

# Chapter 1

## Environment variables

- **PATH:** Is the directory in which executable software are located.

# Chapter 2

## Commands

- **apt:** Abbreviation for *Advanced Package Tool* is a command to simplifies the process of installing, updating and removing softwares packages as well as managing package dependencies. Tha basic syntax of the apt command is:

```
$ sudo apt [Options] command packageName
```

- **cat:** It is a command that displays the content of a text file. Its syntax is:

```
$ cat fileName
```

- **du:** The "Disk Usage" command show the size of file in a directory include subdirectories:

```
$ du -hs filename.ex
```

Usual options:

- **h:** Human-readable.
- **s:** Summary.

- **echo:** It is a command that displays a line of text. Its basic syntax is:

```
$ echo [Options] "Text"
```

We can write a file or append text to existing file with the structures:

```
$ echo "Text" > file.txt  
$ echo "Text" >> file.txt
```

Usual options:

- **n:** Do not output the trailing newline.
- **e:** Enable interpretation of backslash scapes.

To display a internal parameter it can use:

```
$ echo "Text $(parameter)"
```

- **gcc:** Abbreviation for *GNU Compiler Collection* it is a standard compiler on many Unix-like systems. Allows compile in many programming languages such as C, C++, Fortran, etc. Its basic syntax is:

```
$ gcc [Options] sourceFile -o outputFile
```

Usual options:

- **o:** Output file name.
- **wall:** Enable all common warnings.

- **ls:** Abbreviation for *List* list the directory content. Its structure is:

```
$ ls directoryName
```

- **mkdir:** Abbreviation for *Make Directory* is used to create new directories. Its structure is:

```
$ mkdir [Options] directoryName
```

Also, you can remove a directory using:

```
$ rmdir directoryName
```

Usual options:

- **p:** Allows create nested directories:

```
$ mkdir parentDir/sonDir
```

- **mount:** Is a command that attach a filesystem to a directory. When is used, the filesystem become available since the filesystem hierarchy. Its structure is:

```
$ sudo mkdir /mnt/mountPoint
$ mount /dev/deviceName /mnt/mountPoint
```

For see the available devices can use:

```
$ lsblk -l
```

For example:

```
$ mount /dev/deviceName /mnt/mount_name
```

For see the available mount points:

```
$ ls -la /mnt/
```

- **mv:** Move a file between two directories:

```
$ mv /path_file/name_file /destination_path
```

- **pip:** Abbreviation for *Package Installer from Python* is the command that allows access to the **pypi** (*Python Package Index*). With this command you can manage Python Package. Its structure is:

```
$ pip command packageName
```

Some usually commands are:

- **check:** Verify installed packages have compatible dependencies.
- **freeze:** Output installed packages in requirements format.
- **help:** Show help information from pip commands.
- **install-uninstall:** Install or uninstall a package. For install a specific version of a package execute:

```
$ pip install package == version
```

- **list:** Show the list of installed packages.
- **search:** Search for packages.
- **show:** Show information of installed packages.

- **python:** Is the basic command for working in Python. Usually, this command end in the version number of installed Python language, for example **python3**.

Usual options:

- **m:** Allows execute a module from the command line.
- **version:** Show the current Python version.

- **pwd:** Abbreviation for *Print Working Directory* and display the current directory in which the user is working. Options:

- **L:**

- **rm:** Is the abbreviation for *Remove* and allows delete a file. Its structure is:

```
$ rm filename
```

For delete a non-empty directory it can use the options:

- **r:** Recursive, deletes everything inside, including subdirectories.
- **f:** Force.

- **source:** Is used to execute a script in the current shell session. Its structure is:

```
$ source path/script_name.sh
```

- **sudo:** The command sudo is the abbreviation for *Superuser Do* and allows to a user administrative privileges that would otherwise require root access. Is used to install softwares, edit system files and restart services. Have the structure:

```
$ sudo [Option] command [Arguments]
```

- **venv:** Is the abbreviation for *Virtual Environment* and allows create and active a Python virtual environment. The way for create, activate and deactivate a python virtual environment is:

```
$ python3 -m venv EnvironmentName
$ source EnvironmentName/bin/activate
$ deactivate
```

- **vi:** Abbreviation for *Visual* is a screen-oriented text editor that allows create and modify text files. Its syntax is:

```
$ vi fileName
```

# Chapter 3

## Git

- Add a file to a remote repository:

```
$ cp /path/file_name path/repo  
$ git add file_name  
$ git commit -m "Comments"  
$ git push origin main
```

- Change repository visibility:

```
$ gh repo edit --visibility public
```

- Create a new repository:

```
$ mkdir RepoName  
$ cd RepoName  
$ git init  
$ git config --global --add  
safe.directory path/RepoName  
$ echo "Readme file" > README.md  
$ git add .  
$ git commit -m "First commit"  
$ gh repo create GHUser/RepoName  
--public --push --source=.
```

- Login in a remote repository:

```
$ gh auth login
```

# Chapter 4

## Java

- Create and compile a project:

```
$ mkdir -p project_name/src/com/package_name  
$ javac -d ./bin src/com/package_name/*.java  
$ java -cp ./bin com.package_name.file_name
```

# Chapter 5

## Secure Shell (SSH)

- Connect a Raspberry Pi to another device exists two options:

```
$ ssh pi@pi_name.local  
$ ssh pi@pi_IP_address
```

For know the IP address of a Raspberry Pi:

```
$ hostname -I
```

# Chapter 6

## Keyboard shortcuts

- Cancel process:

`ctrl + C`

- Clean the Shell:

`ctrl + L`

- Stop process:

`ctrl + Z`

`fg` for continue.