

# Lab 2: Introduction to Linux and CLI

6COSC002W

Ayman El Hajjar

Week 2

## PLEASE READ ALL OF THIS DOCUMENT

- Linux is more than an OS. It's an idea where everybody grows together and there's something for everybody.
- For this reason, there are many flavours and distributions for Linux.
- The two most commonly used architecture for those distributions are either RPM based or debian based.
  - **RPM** Red Hat Linux and SUSE Linux were the original major distributions that used the .rpm file format, which is today used in several package management systems.
  - **Debian**: Debian is a distribution that emphasizes free software. It supports many hardware platforms. Debian and distributions based on it use the .deb package format[5] and the dpkg package manager and its frontends (such as apt-get or synaptic)
- We will be using Kali and Ubuntu in this module. Both are Debian.

## 1 Learning the shell

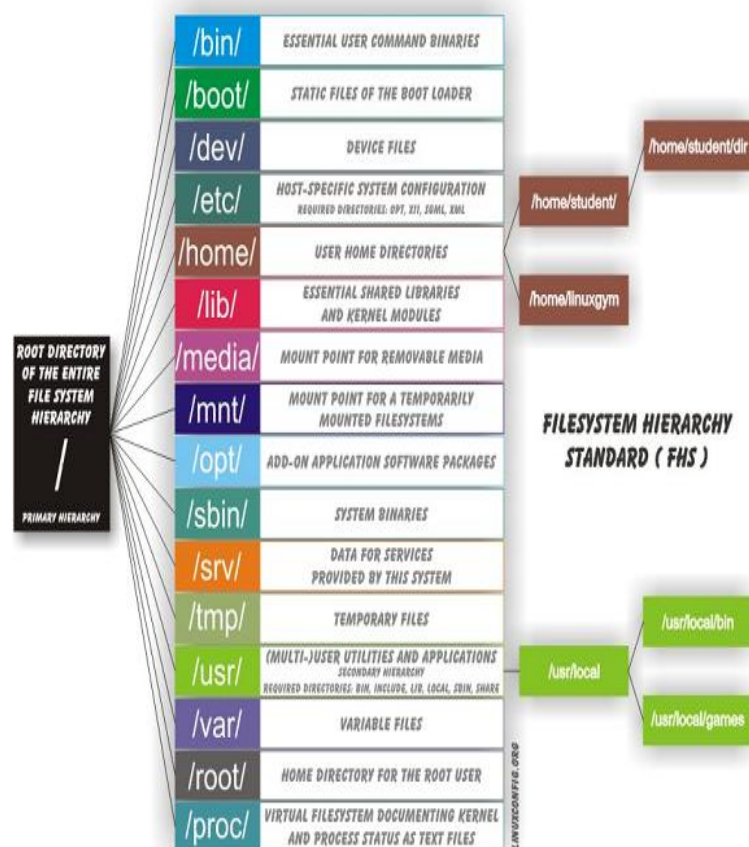
- When we speak of the command line, we are really referring to the shell. The shell is a program that takes keyboard commands and passes them to the operating system to carry out.
- Almost all Linux distributions supply a shell program from the GNU Project called bash.
- When using a graphical user interface, we need another program called a terminal emulator to interact with the shell.
- It is called **terminal**

## 1.1 Simple Commands

- **date**: This command displays the current time and date.
- **cal**: A related command is `cal`, which, by default, displays a calendar of the current month.
- **df**: To see the current amount of free space on your disk drives
- **free**: Likewise, to display the amount of free memory.
- If we press the up-arrow key we can see previous commands.
- **history**: This command remembers the last 500 commands by default in most Linux distributions.
- We can end a terminal session by either closing the terminal emulator window or entering the **exit** command at the shell prompt.

## 1.2 File system tree

- Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a hierarchical directory structure.
- This means that they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories.
- The first directory in the filesystem is called the root directory. The root directory contains files and subdirectories, which contain more files and subdirectories, and so on.
- Note that unlike Windows, which has a separate filesystem tree for each storage device, Unix-like systems such as Linux always have a single filesystem tree, regardless of how many drives or storage devices are attached to the computer.
- Storage devices are attached (or more correctly, mounted) at various points on the tree according to the whims of the system administrator, the person (or persons) responsible for the maintenance of the system.



All files on a Linux system are stored on file systems which are organized into a single inverted tree of directories, known as a file system hierarchy. In the inverted tree, root lies at the top and the branches of directories and sub-directories stretch below the root.

- `\usr`
  - Locally customized software
- `\bin`
  - User commands.
- `\sbin`
  - System administration commands.
- `\etc`
  - Configuration files specific to this system.
- `\etc`
  - Configuration files specific to this system.

- `\var`
  - Files that dynamically change (e.g. databases, cache directories, log files, printer spooled documents, and website content) may be found under `/var`.
- `\home`
  - Home directories where regular users store their personal data and configuration files.
- `\root`
  - Purpose
    - \* Home directory for the administrative superuser, root.
- `\tmp`
  - Purpose
    - \* A world-writable space for temporary files. Files more than 10 days are automatically be deleted from that directory.

### 1.3 Commands to navigate file systems

#### Commands options

- Most commands use options consisting of a single character preceded by a dash, such as `-l`.
  - But many commands, including those from the GNU Project, also support long options, consisting of a word preceded by two dashes.
  - If you want to know what options you can use for specific commands, you can read the manual for each command.
  - **man**: If you type **man ls** for example, this will give you the manual for `ls` command including all options for `ls`.
- 
- **pwd**: This command displays the current working directory.
    - When you are using GIU, you can easily identify in which directory you are working.
    - When you using command line, you need to use a command to help you identify this.
  - **ls**: lists the files and directories in the current working. directory.
    - **ls -a** or **ls --all** List all files, even those with names that begin with a period, which are normally not listed (i.e., hidden).

- **-d --directory** Changes the working directory to the previous working directory.
- **-F --classify** This option will append an indicator character to the end of each listed name (for example, a forward slash if the name is a directory).
- **-h --human-readable** In long format listings, display file sizes in human-readable format rather than in bytes.
- **-l** display its results in long format. This format contains a great deal of useful information.
- **cd**: To change your working directory (where we are standing in our tree-shaped maze) we use the **cd** command: Type **cd** followed by the pathname of the desired working directory
  - **cd /bin**
  - This will change your working directory to **bin** directory
  - There are some shortcuts for the **cd** command
    - \* **cd** Changes the working directory to your home directory.
    - \* **cd -** Changes the working directory to the previous working directory.
    - \* **cd ..** move to the parent of the current working directory.
    - \* **cd ~** Changes the working directory to the home directory of username.
- **file**: We will use the **file** command to determine a file's type.
- **less**: The **less** command is a program to view text files.
  - Once started, the **less** program allows you to scroll forward and backward through a text file. For example, to examine the file that defines all the system's user accounts, enter the following command:
    - \* **less /etc/passwd**

## 2 Manipulating Files and Directories

- **mkdir**: The **mkdir** command is used to create directories.
  - **mkdir lab1** will create a directory called **lab1**
  - type **man mkdir** to learn about the different options of the **mkdir**
- **cp**: The **cp** command copies files or directories.
  - **cp item1 item2** will copy the single file or directory **item1** to file or directory **item2**
  - type **man cp** to learn about the different options of the **cp** command

- **mv**: The mv command performs both file moving and file renaming, depending on how it is used.
  - **mv item1 item2** move or rename file or directory item1 to item2
  - type **man mv** to learn about the different options of the **mv** command
- **rm**: The rm command is used to remove (delete) files and directories.
  - **rm item1** will remove item1 file or directory item1 (if empty)
  - type **man rm** to learn about the different options of the **rm** command and how to delete a directory even if it contains documents.

### 3 Permission

- Operating systems in the Unix tradition differ from those in the MS-DOS tradition in that they are not only multitasking systems but also multiuser systems.
- What exactly does this mean? It means that more than one person can use the computer at the same time. While a typical computer will likely have only one keyboard and monitor, it can still be used by more than one user. For example, if a computer is attached to a network or the Internet, remote users can log in via ssh (secure shell) and operate the computer. In fact, remote users can execute graphical applications and have the graphical output appear on a remote display. The X Window System supports this as part of its basic design.
- The multiuser capability of Linux is not a recent “innovation” but rather a feature that is deeply embedded into the design of the operating system. Considering the environment in which Unix was created, this makes perfect sense. Years ago, before computers were “personal,” they were large, expensive, and centralized. A typical university computer system, for example, consisted of a large central computer located in one building and terminals located throughout the campus, each connected to the large central computer. The computer would support many users at the same time.
- In order to make this practical, a method had to be devised to protect the users from each other. After all, the actions of one user could not be allowed to crash the computer, nor could one user interfere with the files belonging to another user
- Access rights to files and directories are defined in terms of read access, write access, and execution access. If we look at the output of the ls command, we can get some clue as to how this is implemented.
  - **mkdir lab1** will create a directory called lab1
  - type **man mkdir** to learn about the different options of the **mkdir**



- 1 File type
- 2 Owner permissions
- 3 Group permissions
- 4 World permissions

Attribute	Files	Directories
r	Allows a file to be opened and read.	Allows a directory's contents to be listed if the execute attribute is also set.
w	Allows a file to be written to or truncated; however, this attribute does not allow files to be renamed or deleted. The ability to delete or rename files is determined by directory attributes.	Allows files within a directory to be created, deleted, and renamed if the execute attribute is also set.
x	Allows a file to be treated as a pro-gram and executed. Program files written in scripting languages must also be set as readable to be executed.	Allows a directory to be entered; e.g., <i>cd directory</i> .

- The remaining nine characters of the file attributes, called the file mode, represent the read, write, and execute permissions for the file's owner, the file's group owner, and everybody else.
- When set, the r, w, and x mode attributes have certain effects on files and directories as below:

## chmod - Change file mode

- To change the mode (permissions) of a file or directory, the chmod command is used. Be aware that only the file's owner or the superuser can change the mode of a file or directory. chmod supports two distinct ways of specifying mode changes: octal number representation and symbolic representation. We will cover octal number representation first.
  - Directories have directory permissions. The directory permissions restrict different actions than with files or device nodes.

Permission	Action	chmod option
read	(view contents, i.e. ls command)	r or 4
write	(create or remove files from dir)	w or 2
execute	(cd into directory)	x or 1

- To give file permission for a file called filename for example in this specific order execute for owner , read & write for group and read for other, we use the command as below:

- **chmod 764**
- Where 7 is read write execute  $4 + 2 + 1 = 7$
- Where 6 is read write  $4 + 2 = 6$
- Where 4 is read 4

### Super user commands

- One of the recurrent problems for regular users is how to perform certain tasks that require superuser privileges.
- These tasks include installing and updating software, editing system configuration files, and accessing devices.
- In the Windows world, this is often done by giving users administrative privileges. This allows users to perform these tasks. However, it also enables programs executed by the user to have the same abilities. This is desirable in most cases, but it also permits malware (malicious software) such as viruses to have free run of the computer.
- To be able to use commands that requires higher privilege you should use **sudo** command
- You will be asked for a password.
  - In linux when you enter a password in the terminal, nothing shows up. This is done on purpose in order to hide the length of the password.
  - If for example I want to copy a file called run.sh to bin directory, I will need root privilege.
  - To do this I type **sudo cp run.sh /bin**



## Networking

- We need to know the IP address of our machine.
- You can use **ifconfig** command or **ip address** or **ip a** command to do so.
  - This will list all your network interfaces and the different IP address allocated for each interface. Interfaces can be for example a network card, a wireless network card, or a virtual network card.
  - if ifconfig doesn't work and you get an error to install the command, you can type **sudo apt-get install net-tools** to install it.
- If I want to bring down a specific interface (net1 for example) I can type
  - **ifconfig net1 down**
- If I want to bring up a specific interface (net1 for example) I can type
  - **ifconfig net1 up**
- If your interface was not allocated an IP address dynamically you can set it up yourself. For example if I want to give IP 192.168.10.2 for the interface net1 I use this command
  - **ifconfig net1 192.168.10.2 netmask 255.255.2550**
- If I want to check connectivity with another device on my network I use the command **ping**
  - **ping 192.168.10.4** will try to connect using ICMP protocol with this ip and see if this IP is reachable.