Operating Systems 1  
Assignment X

RAJ PATIL  
CS18BTECH11039

# The Linux Foundation course – Chapter 1:9

**PLAGIARISM STATEMENT**

I certify that this assignment is my own work, based on my personal  
study and research and that I have acknowledged all materials and sources  
used in its preparation, whether they be books, articles, reports, lecture notes,  
and any other kind of document, electronic or personal communication. I also  
certify that this assignment has not previously been submitted for  
assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

**Name: Raj Patil**  
**Date: 2nd December 2019**  **Signature** 

Chapter 1

- talks about the Linux foundation

- its subsidiary open-source endeavours in different fields

- talks about three major families of Linux distributions (referred to as distros from here onwards)

- The Red Hat family

- the SUSE family

- the Debian family

- a mention is made about the primary difference between these which are limited to the package managers they use, the locations where specific files are stored, and that's that for the significant differences.

Chapter 2

The second chapter focuses on the abstract concepts and philosophies Linux is based on.

The first segment of the chapter focus on the history of Linux where it gives us an idea of how ubiquitous Linux is - in web servers, phones, embedded systems and how it all began.

In the next segment, the underlying ideology of Linux is discussed: everything (processes, devices and files) can be represented as files which it borrows heavily from the UNIX OS ( designed for more powerful computers than a PC). All files are stored hierarchically with root being the top node of the system. It then introduces what daemons (processes managing other processes in background in the crudest sense) are.

We are then introduced how Linux is developed and the cause of its rapid development:- ( a significant kernel release occurs every 2 or 3 months)

- it's updated in patches- pushed in by developers which can be

- changes

- additions

- subtractions

- .. to the existing kernel

In the last section of this chapter, basic Linux terminology is clarified.

Chapter 3

The Linux boot process is discussed in this chapter. The reason to address this is mentioned as to help the user when troubleshooting problems and tweaking the computer's performance as needed.

The primary stages of control flow are as follows(described alongside if need be):

1. power ON

2. BIOS

- Basic input/output system

- initialises the hardware

- tests the main memory

- tests the screen and keyboard

- process also called POST:

- Power on self test

- BIOS software stored on the ROM on motherboard

3. MBR --> boot loader

- control passed on to MBR after POST

- MBR: identifies the bootable partition on the device memory and passes on command to the boot loader

- size of just 512 Bytes

4. Boot Loader

- boot loader stored on a certain sector of the device memory

- vital info such as date,time and peripherals of prominence are loaded:

- from the CMOS values

- complementary metal oxide semiconductor

- name of the tech used to maintain this data

- some common boot loaders for linux:

- GRUB (GRand Unified Boot loader)

- ISOLINUX ( booting from peripheral storages)

- DAS U-Boot ( for embedded devices )

- boot loader has the resposibility to

- load the kernel image

- load the initial RAM disk/filesystem

- .. into the memory

5. Kernel

- initialises computer's memory

- configures all hardware attached to the system

- also loads some necessary user-space processes

6. Initial RAM disk

- initramfs filesystem:

- contains programs and binary files

- reponsible for mounting proper root filesystem like:

- providing kernel functionality

- providing device drivers for mass storage controllers ( using udev )

- udev: resposible for figuring out which devices are present

- locating device drivers they need to operate properly

- loading these drivers

- upon succeeding, initramfs is cleared from the RAM

- /sbin/init is executed

7. init ( parent process initiated)(parent of all non-kernel processes)

- hands-on the mounting and pivoting to the final filesystem

- Prompts text logins of no GUI being used else...

8. X windows system ( the GUI )

Partitions and file systems:

- partition: physically contiguous section of a disk

- filesystem: method of storing/retrieving files on/from a hard disk

- FHS(file system heirarchy):

- files stored according to this standard layout

- filesystem names are case sensitive

- tree : for viewing the heirarchy in command line

The next part of the chapter guides us through choosing the right Linux distribution according to our needs and walks us through an install.

Chapter 4

This chapter focuses on the functionalities captured by way of using the GUI(graphic user interface).

GNOME is a popular GUI choice for Linux Operating systems.

-its default display manager is called gdm(gnome display manager).

-later on, it describes some basic behaviour of the OS: for instance: suspension of the leading to sleep.

Then we go on talking about file placement via the file explorer:

-Every user has its own home directory in the system.

-Places menu: allows to access to different parts of the computer and the network.

-Nautilus, now known as GNOME files, is the official file manager for the GNOME desktop.

-later on, we explore customising the environment a bit using themes.

Chapter 5

This chapter focuses on configuring the system using the GUI.

Topics touched:

-basic configuration through the system settings panel

-timekeeping via the system settings window

-a bit about the network time protocol

-tweaking the resolution using the display panel

-Network manager functionality:

-wireless networks

-VPNs

-password management

-high-level package management systems in different families of distributions:

- dpkg for the Debian family

- rpm for the red-hat family and the open-SUSE family

Chapter 6

The main focus of this one was to introduce the typical user-level applications such as text editors, web browsers, email clients, video and audio manipulation applications, productivity tools and so on.

Not going to state all of them, just stating what I use:

- vim as my text-editor with tmux to get a sort of snappy light-weight IDE, shifting to neovim soon

- lynx for quick searches from the CLI and firefox for daily-browsing

- lynx in unison with pipelined pandoc for quick file conversions from say markdown to commonly renderable formats like HTML.

- GIMP for editing images: I blog hence need to optimise images being uploaded on the web; chose GIMP over photoshop because it's free and copy-left as opposed to paying for copyright software.

- GCC ( itself and through g++) for compiling and linking source files.

- gdb for debugging

- valgrind for detecting memory leaks.

- VLC for a movie once in a while.

- used to use VS-code, but vim with some plugin customisation does the job well and haven't used it now for 2 months.

Chapter 7

this chapter focuses on the basic functionality accessible via the CLI

Initially, a basic intro to the terminal is given, and actions like shutting down the computer and rebooting are discussed.

Later, they mention pathnames

- relative ( . , .. for instance )

- relative pathnames are relative from the present working directory: displayable quickly using the pwd command.

- absolute (begins with the root directory and follows the tree)

- then basic navigation commands through the file system are discussed:

- cd: change directory with attributes like . , .. , - and pathnames directly..

- ls: to list the current working directories contents

- tree: for an overview of the files and with -d to just see the output for directories

- mkdir : to create directories

- rmdir and rm -r: to remove directories

- rm on its own to delete files

- attributes of rm such as i,f,r

- Then we move on to searching within the file system using the wild card characters \*,? and some more with the following commands:

- locate: search in the database to find all file names matching a pattern

- find: locate files recursively from a given set of directories

- its exec attribute to perform some actions on the found files

- touch's functionality is then discussed, including but not limited to:

- creating placeholder files quickly

- timestamping files as you desire along with accessing, changing or editing times of the file

- then we talk about upgrading the packages using a package management tool such as apt on ubuntu:

- what I do personally is the following 3 commands in succession:

- sudo apt-get update : updates dependencies

- sudo apt-get upgrade : downloads and installs the new dependencies

- sudo apt-get autoremove : cleans up the old packages which are not required anymore

Chapter 8

This chapter focuses on how to seek help regarding the working of functionalities provided by the OS.

4 primary sources are discussed:

- The man pages: searches, formats and displays the pages

- GNU info system: a system of browsable linked nodes of info about commands: ideal for an interactive movement between related commands

- help: shorter descriptions accessible by passing -h as an argument to the command

- online documentation for the OS

Chapter 9

This was by far the most interesting chapter where I actually learned something new and of value.

It talks about processes, introducing the notion of nice-values and priorities(inversely related) for prioritising tasks.

- basic concepts discussed in the class ( pid, ppid, tid and so on) are not presented here

- ps: used to obtain a static overview of the current processes

- top: for a dynamic overview(refreshed every 2 seconds) of the same

- for more interactability and info, one could use htop or atop

- was introduced to the load average metric to gauge CPU utilisation in the close past

- Linux supporting background (daemons) and foreground processes was then discussed

- then scheduling tasks to be performed in the future was discussed:

- at for a non-interactive command once at a specified time

- cron ( from Chronos: the Greek personification of time) for periodically scheduled tasks: the schedule is stored in crontab