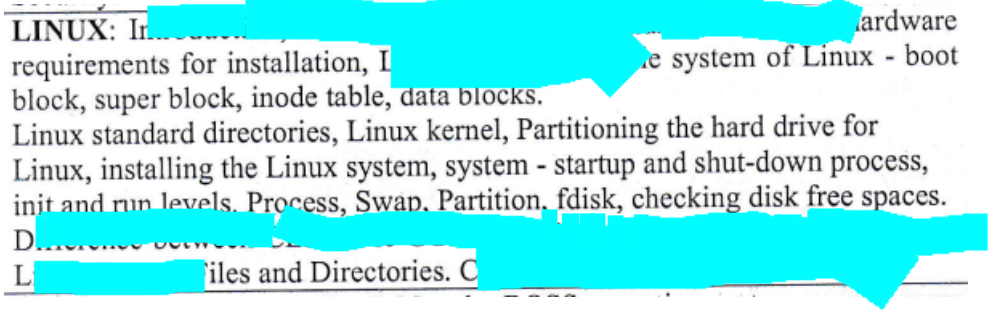

LINUX: I[redacted]ardware
requirements for installation, I[redacted]e system of Linux - boot
block, super block, inode table, data blocks.
Linux standard directories, Linux kernel, Partitioning the hard drive for
Linux, installing the Linux system, system - startup and shut-down process,
init and run levels. Process, Swap, Partition, fdisk, checking disk free spaces.
D[redacted]
L[redacted]iles and Directories. C[redacted]

What Is Linux

Linux is an open-source operating system like other operating systems such as Microsoft Windows

, Apple Mac OS, iOS, Google android, etc. An operating system is a software that enables the communication between computer hardware and software. It conveys input to get processed by the processor and brings output to the hardware to display it. This is the basic function of an operating system. Although it performs many other important tasks, let's not talk about that.

Linux is around us since the mid-90s. It can be used from wristwatches to supercomputers. It is everywhere in our phones, laptops, PCs, cars and even in refrigerators. It is very much famous among developers and normal computer users.

HISTORY OF LINUX

Evolution of Unix

In 1969, a team of developers of Bell Labs started a project to make a common software for all the computers and named it as 'Unix'. It was simple and elegant, used 'C' language instead of assembly language and its code was recyclable. As it was recyclable, a part of its code now commonly called 'kernel' was used to develop the operating system and other functions and could be used on different systems. Also its source code was open source.

Initially, Unix was only found in large organizations like government, university, or larger financial corporations with mainframes and minicomputers (PC is a microcomputer).

Unix Expansion

In eighties, many organizations like IBM, HP and dozen other companies started creating their own Unix. It result in a mess of Unix dialects. Then in 1983, Richard Stallman developed GNU project with the goal to make it freely available Unix like operating system and to be used by everyone. But his project failed in gaining popularity. Many other Unix like operating system came into existence but none of them was able to gain popularity.

Evolution of Linux

In 1991, Linus Torvalds a student at the university of Helsinki, Finland, thought to have a freely available academic version of Unix started writing its own code. Later this project became the Linux kernel. He wrote this program specially for his own PC as he wanted to use Unix 386 Intel computer but couldn't afford it. He did it on MINIX using GNU C compiler. GNU C compiler is still the main choice to compile Linux code but other compilers are also used like Intel C compiler.

He started it just for fun but ended up with such a large project. Firstly he wanted to name it as 'Freax' but later it became 'Linux'.

He published the Linux kernel under his own license and was restricted to use as commercially. Linux uses most of its tools from GNU software and are under GNU copyright. In 1992, he released the kernel under GNU General Public License.

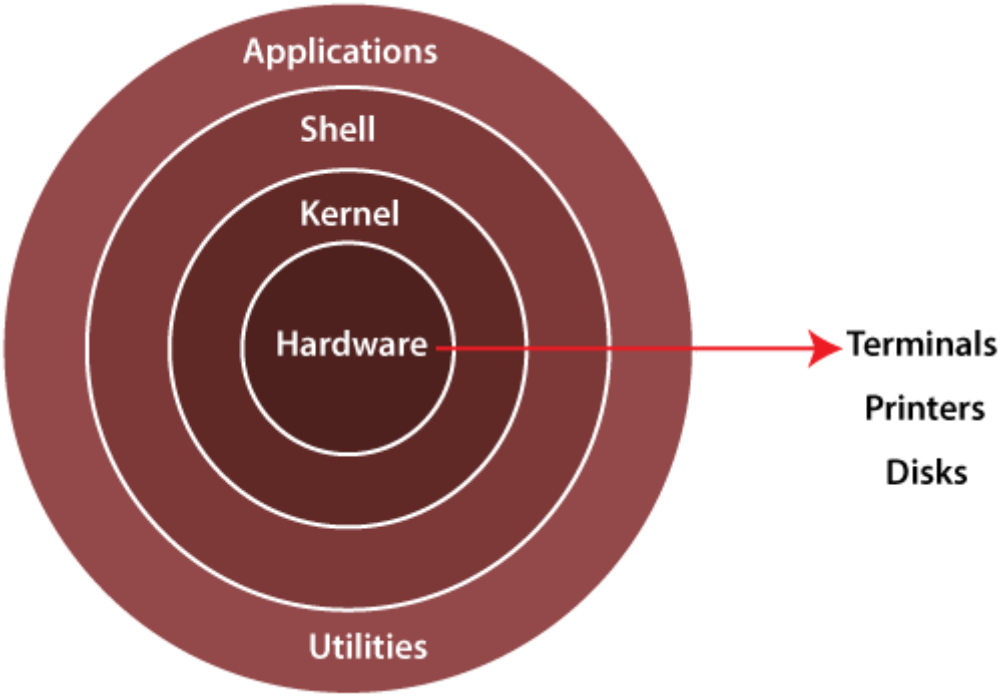
Linux Today

Today, supercomputers, smart phones, desktop, web servers, tablet, laptops and home appliances like washing machines, DVD players, routers, modems, cars, refrigerators, etc use Linux OS.

Linux Features

- **Multiuser capability:** Multiple users can access the same system resources like memory, hard disk, etc. But they have to use different terminals to operate.
- **Multitasking:** More than one function can be performed simultaneously by dividing the CPU time intelligently.
- **Portability:** Portability doesn't mean it is smaller in file size or can be carried in pen drives or memory cards. It means that it support different types of hardware.
- **Security:** It provides security in three ways namely authenticating (by assigning password and login ID), authorization (by assigning permission to read, write and execute) and encryption (converts file into an unreadable format).
- **Live CD/USB:** Almost all Linux distros provide live CD/USB so that users can run/try it without installing it.
- **Graphical User Interface (X Window system):** Linux is command line based OS but it can be converted to GUI based by installing packages.
- **Support's customized keyboard:** As it is used worldwide, hence supports different languages keyboards.
- **Application support:** It has its own software repository from where users can download and install many applications.
- **File System:** Provides hierarchical file system in which files and directories are arranged.
- **Open Source:** Linux code is freely available to all and is a community based development project.

Architecture of Linux system



The Linux operating system's architecture mainly contains some of the components: **the Kernel, System Library, Hardware layer, System,** and **Shell utility**.

1. Kernel:- The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS. This operating system contains distinct types of modules and cooperates with underlying hardware directly. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

- Monolithic Kernel
- Micro kernels
- Exo kernels
- Hybrid kernels

2. System Libraries:- These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

3. System Utility Programs:- It is responsible for doing specialized level and individual activities.

4. Hardware layer:- Linux operating system contains a hardware layer that consists of several peripheral devices like CPU, HDD, and RAM.

5. Shell:- It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are the **graphical shells** and **command-line shells**.

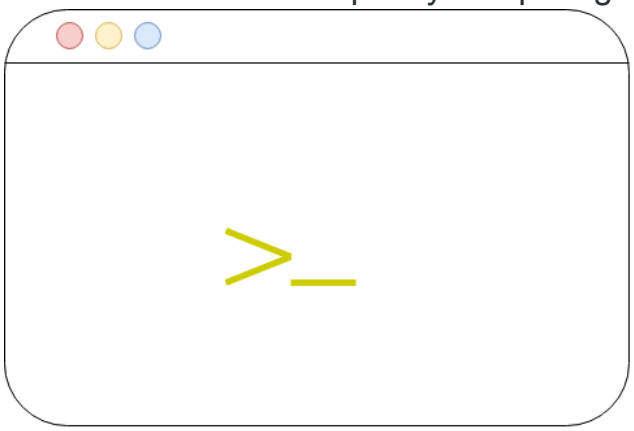
The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.

There are a few types of these shells which are categorized as follows:

- Korn shell
- Bourne shell
- C shell
- POSIX shell

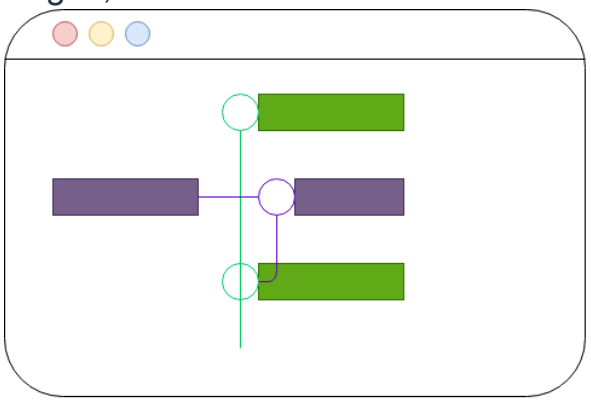
Difference between CLI and GUI

CLI is the word form used for **Command Line Interface.** CLI permits users to put in writing commands associate degree exceedingly in terminal or console window to interact with an operating system. CLI is a platform or medium wherever users answer a visible prompt by writing a command and get the response from the system, for this users have to be compelled to kind command or train of command for performing the task. CLI is suitable for pricey computing wherever input exactitude is the priority.



CLI

GUI stands for **Graphical User Interface.** GUI permits users to use the graphics to interact with an operating system. In the graphical user interface, menus are provided such as windows, scrollbars, buttons, wizards, painting pictures, alternative icons, etc. It's intuitive, simple to find out, and reduces psychological feature load. In GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc.



GUI

Let's see the difference between GUI and CLI:

S.NO	CLI	GUI
1.	CLI is difficult to use.	Whereas it is easy to use.
2.	It consumes low memory.	While consuming more memory.
3.	In CLI we can obtain high precision.	While in it, low precision is obtained.
4.	CLI is faster than GUI.	The speed of GUI is slower than CLI.
5.	CLI operating system needs only a keyboard.	While GUI operating system needs both a mouse and keyboard.
6.	CLI's appearance can not be modified or changed.	While its appearance can be modified or changed.
7.	In CLI, input is entered only at a command prompt.	While in GUI, the input can be entered anywhere on the screen.
8.	In CLI, the information is shown or presented to the user in plain text and files.	While in GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc.
9.	In CLI, there are no menus provided.	While in GUI, menus are provided.
10.	There are no graphics in CLI.	While in GUI, graphics are used.
11.	CLI do not use any pointing devices.	While it uses pointing devices for selecting and choosing items.
12.	In CLI, spelling mistakes and typing errors are not avoided.	Whereas in GUI, spelling mistakes and typing errors are avoided.
13.	Some command-line environments provide multitasking but it is complicated to see several things on one screen.	GUI enables a user to easily observe and operate various things at once.
14.	CLI enables a user to simply script a series of instructions to carry out a task or execute a program.	GUI does not provide the facility to script a sequence of commands.

Difference between Linux and Windows

Linux: Linux could be a free and open supply OS supported operating system standards. It provides programming interface still as programme compatible with operating system primarily based systems and provides giant selection applications. A UNIX operating system additionally contains several severally developed parts, leading to UNIX operating system that is totally compatible and free from proprietary code. **Windows:** Windows may be a commissioned OS within which ASCII text file is inaccessible. it's designed for the people with the angle of getting no programming information and for business and alternative industrial users. it's terribly straightforward and simple to use. The distinction between Linux and Windows package is that Linux is completely freed from price whereas windows is marketable package and is expensive. Associate operating system could be a program meant to regulate the pc or computer hardware Associate behave as an treater between user and hardware. Linux is a open supply package wherever users will access the ASCII text file and might improve the code victimisation the system. On the opposite hand, in windows, users can't access ASCII text file, and it's a authorized OS. Let's see that the difference between Linux and windows:

S.NO	Linux	Windows
1.	Linux is a open source operating system.	While windows are the not the open source operating system.
2.	Linux is free of cost.	While it is costly.
3.	It’s file name case-sensitive.	While it’s file name is case-insensitive.
4.	In linux, monolithic kernel is used.	While in this, micro kernel is used.
5.	Linux is more efficient in comparison of windows.	While windows are less efficient.
6.	There is forward slash is used for Separating the directories.	While there is back slash is used for Separating the directories.
7.	Linux provides more security than windows.	While it provides less security than linux.
8.	Linux is widely used in hacking purpose based systems.	While windows does not provide much efficiency in hacking.
9.	There are 3 types of user account – (1) Regular , (2) Root , (3) Service account	There are 4 types of user account – (1) Administrator , (2) Standard , (3) Child , (4) Guest
10.	Root user is the super user and has all administrative privileges.	Administrator user has all administrative privileges of computers.
11.	Linux file naming convention in case sensitive. Thus, sample and SAMPLE are 2 different files in Linux/Unix operating system.	In Windows, you cannot have 2 files with the same name in the same folder.

The Linux Kernel

The main purpose of a computer is to run a *predefined sequence of instructions*, known as a **program**. A program under execution is often referred to as a **process**. Now, most special purpose computers are meant to run a single process, but in a sophisticated system such a general purpose computer, are intended to run many processes simultaneously. Any kind of process requires hardware resources such as Memory, Processor time, Storage space, etc. In a General Purpose Computer running many processes simultaneously, we need a middle layer to manage the distribution of the hardware resources of the computer efficiently and fairly among all the various processes running on the computer. This middle layer is referred to as the **kernel**. Basically the kernel virtualizes the common hardware resources of the computer to provide each process with its own virtual resources. This makes the process seem as it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes. This schematically represented below:

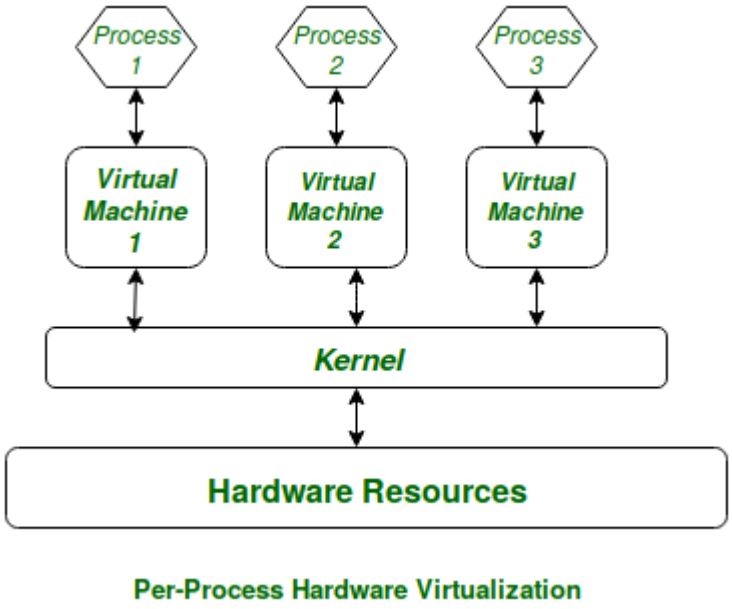


Figure: Virtual Resources for each Process

The **Core Subsystems** of the **Linux Kernel** are as follows:

1. The Process Scheduler
2. The Memory Management Unit (MMU)
3. The Virtual File System (VFS)
4. The Networking Unit
5. Inter-Process Communication Unit

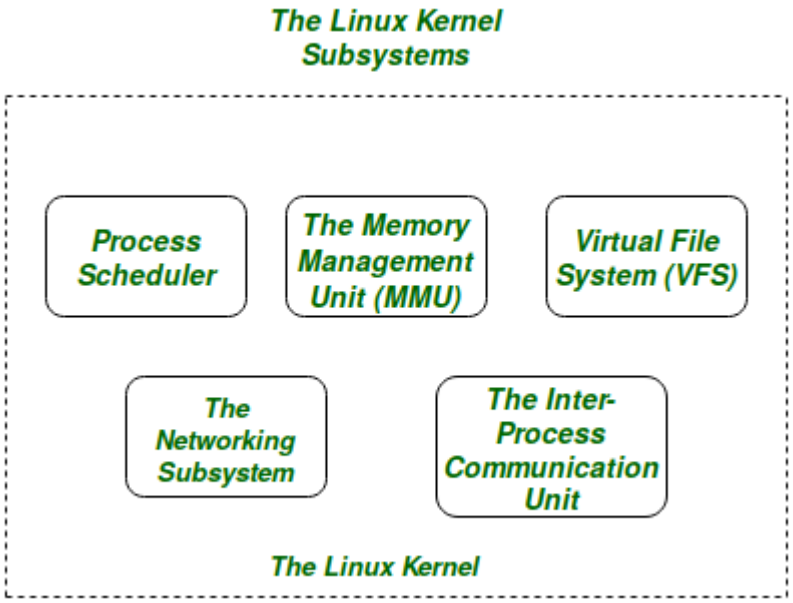


Figure: The Linux KernelFor the purpose of this article we will only be focussing on the 1st three important subsystems of the Linux Kernel. The basic functioning of each of the 1st three subsystems is elaborated below:

- **The Process Scheduler:** This kernel subsystem is responsible for fairly distributing the CPU time among all the processes running on the system simultaneously.
- **The Memory Management Unit:** This kernel sub-unit is responsible for proper distribution of the memory resources among the various processes running on the system. The MMU does more than just simply provide separate virtual address spaces for each of the processes.
- **The Virtual File System:** This subsystem is responsible for providing a unified interface to access stored data across different filesystems and physical storage media.

Introduction to Open-Source and its benefits

What does Open-source mean?

The term **Open-source** is closely related to Open-source software (OSS).
Open-source software is a type of computer software that is released under a license, but the source code is made available to all the users. The copyright holders of such software allow the users to use it and do some valuable modifications in its source code to add some new features, to improve the existing features, and to fix bugs if there are any. Because of this reason only Open-source software is mostly developed collaboratively.

Some famous examples of Open-source products are :

- **Operating systems –**
  Android, Ubuntu, Linux
- **Internet browsers –**
  Mozilla Firefox, Chromium
- **Integrated Development Environment (IDEs) –**
  Vs code (Visual Studio Code), Android Studio, PyCharm, Xcode

Open-source community and Contributions :

The **open-source community** is a worldwide community of programmers and software developers who are continuously working on various open-source projects to make our lives better. This community is self-governing and self-organizing, there are no executives to take the decisions solely. This community plays a very crucial role in the sustainability of various open-source organizations.

The contributions made in any open-source project which improves its usability are called **open-source contributions**. These contributions can be of any form not only some software codes like we can work on improving its **documentation**, improving its **UI/UX (user interface and design)**, organize meetups, or find new collaborators.

Benefits of Open-source contributions :

- We code for real-world open-source projects.
- It refines our existing knowledge of programming and also helps us to learn new skills.
- Many open-source projects offer mentorship programs to guide and help us through our first few contributions.
- We need not develop the whole thing from scratch, we just have to fork our favorite projects and start experimenting with them.
- After making any open-source contribution, we get immediate feedback regarding our developmental work.
- While doing open-source contributions, we interact with like-minded developers from all over the world and build connections along the way.
- As we get more closer to the open-source community, we get to know much more about our field of interest and other related fields.
- The most important aspect of open-source contributions is It may fetch us a job in our field of interest.

Hence, a large number of students are heading towards open-source contributions because these days quality open-source contributions in some good projects are seen as an alternative to the good internships with developer’s profile. The reason is we get the same kind of exposure to work and learn the required skills for software development like any good internships at Google or Microsoft.