1.1 Features of Open Source Operating Systems, Core Linux Distributions, Architecture, OS Services, System Calls, Run Levels.

Features of Open Source Operating Systems

The term **"open source"** refers to computer software or applications where the owners or copyright holders enable the users or third parties to use, see, and edit the product's source code. The source code of an open-source OS is publicly visible and editable. The usually operating systems such as Apple's iOS, Microsoft's Windows, and Apple's Mac OS are closed operating systems. Open-Source Software is licensed in such a way that it is permissible to produce as many copies as you want and to use them wherever you like. It generally uses fewer resources than its commercial counterpart because it lacks any code for licensing, promoting other products, authentication, attaching advertisements, etc.

The open-source operating system allows the use of code that is freely distributed and available to anyone and for commercial purposes. Being an open-source application or program, the program source code of an open-source OS is available. The user may modify or change those codes and develop new applications according to the user requirement. Some basic examples of the open-source operating systems are **Linux, Open Solaris, Free RTOS, Open BDS, Free BSD, Minix,** etc.

In **1997**, the first Open-Source software was released. Despite the industry, there are now Open-Source alternatives for every Software program. Thanks to technological developments and innovations, many Open-Source Operating Systems have been developed since the dawn of the **21st** century.

Advantages and Disadvantages of Open-Source Operating System:

Advantages

1. Reliable and efficient
   The open-source operating systems are most reliable and efficient. Thousands of eyes monitor these because the source code is public. As a result, if there are any bugs or errors, they are fixed by the best developers worldwide.
2. Cost-efficient
   Most of the open-source operating systems are free. And some of them are far less expensive than commercially closed products.
3. Flexibility
   The great advantage is you may customize it as per your requirement. And there is creative freedom.

Disadvantages

1. Complicated
   It is not as user-friendly as the ones that are closed. To use this software, you must have a basic understanding of technology.
2. Security risk
   Despite the defects having been detected, there is a risk of assaults because the attackers have access to the source code.
3. No support
   If you run across an issue, there is no customer support available to assist you.

Linux Operating System

Introduction

- Linux is a community of open-source Unix like operating systems that are based on the Linux Kernel.
- It was initially released by Linus Torvalds on September 17, 1991.
  It is a free and open-source operating system and the source code can be modified

and distributed to anyone commercially or non-commercially under the GNU General Public License.

- Initially, Linux was created for personal computers and gradually it was used in other machines like servers, mainframe computers, supercomputers, etc. Nowadays, Linux is also used in embedded systems like routers, automation controls, televisions, digital video recorders, video game consoles, smartwatches, etc.
- The biggest success of Linux is Android (2005, operating system Google released in 2008) it is based on the Linux kernel that is running on smartphones and tablets.
- Due to android Linux has the largest installed base of all general-purpose operating systems. Linux is generally packaged in a Linux distribution.

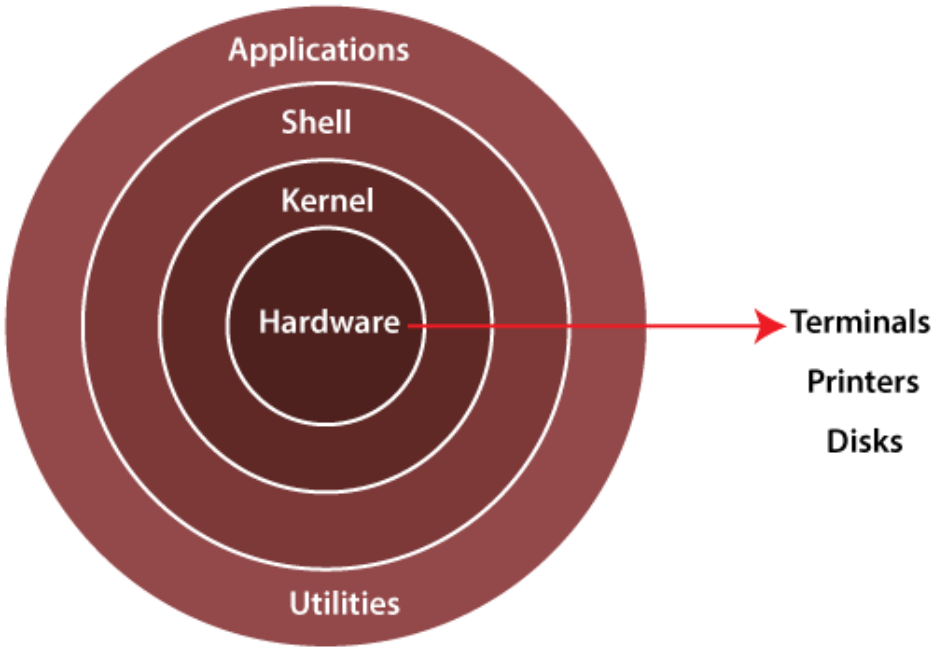
Linux Distributions

It is an OS that is composed of a software-based collection on Linux kernel or we can say the distribution includes the Linux Kernel. It is supporting software and libraries. We can obtain Linux-based OS by downloading any Linux distribution. These types of distributions exists for distinct types of devices such as **personal computers, embedded devices**, etc. Around more than 600 Linux distributions are existed and a few of the famous Linux distributions are listed as follows:

- Deepin
- OpenSUSE
- Fedora
- Solus
- Debian
- Ubuntu
- Elementary
- Linux Mint
- Manjaro

- MX Linux

Architecture of Linux Operating 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

Operating System Processes

An Operating System provides services to both the users and to the programs.
It provides programs an environment to execute.
It provides users the services to execute the programs in a convenient manner.
Following are a few common services provided by an operating system −

- Program execution
- I/O operations
- File System manipulation
- Communication

- Error Detection
- Resource Allocation
- Protection

Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated as a process.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management −
Loads a program into memory.
Executes the program.
Handles program's execution.
Provides a mechanism for process synchronization.
Provides a mechanism for process communication.
Provides a mechanism for deadlock handling.

I/O Operation

An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.

An Operating System manages the communication between user and device drivers.

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

File system manipulation

A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Following are the major activities of an operating system with respect to file management −

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

Communication

In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes. Multiple processes communicate with one another through communication lines in the network.

The OS handles routing and connection strategies, and the problems of contention and security. Following are the major activities of an operating system with respect to communication −

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.
-

Error handling

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling −

- The OS constantly checks for possible errors.

- The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management −

- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.

Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection −

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

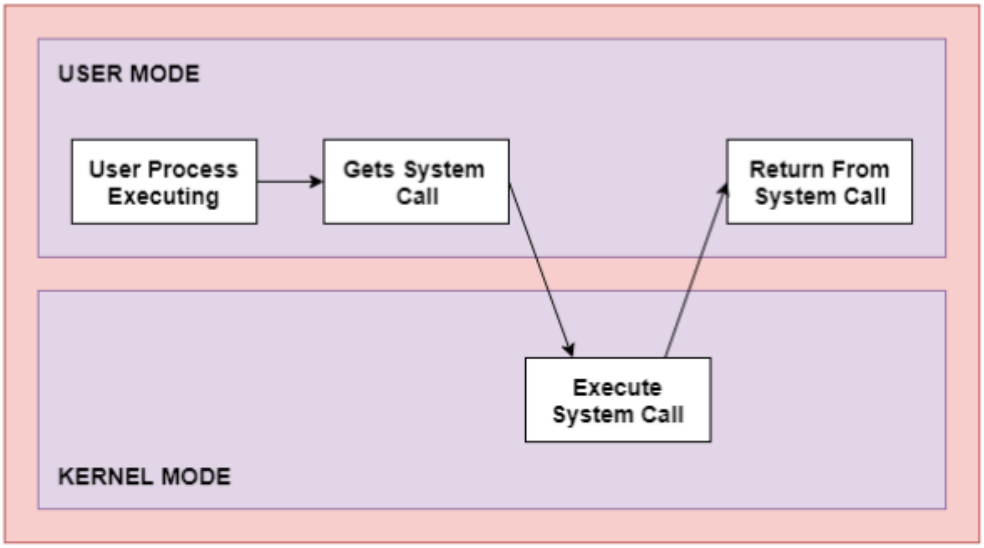
System Calls

In computing, a **system call** is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. A system call is a way for programs to **interact with the operating system**. A computer program makes a system call when it makes a request to the operating system’s kernel. System call **provides** the services of the operating system to the user programs via Application

Program Interface(API). It provides an interface between a process and operating system to allow user-level processes to request services of the operating system. System calls are the only entry points into the kernel system. All programs needing resources must use system calls.

Services Provided by System Calls :

1. Process creation and management
2. Main memory management
3. File Access, Directory and File system management
4. Device handling(I/O)
5. Protection
6. Networking, etc.



Types of System Calls : There are 5 different categories of system calls –

1. Process Control
   These system calls deal with processes such as process creation, process termination etc.

2. File Management
   These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.
3. Device Management
   These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.
4. Information Maintenance
   These system calls handle information and its transfer between the operating system and the user program.
5. Communication
   These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

Examples of Windows and Unix System Calls –

System Calls in windows and unix

Process	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	Fork() Exit() Wait()
File Manipulation	CreateFile() ReadFile() WriteFile ()CloseHandle()	Open() Read() Write() Close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	Ioctl() Read() Write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	Getpid()Alarm()Sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	Pipe() Shmget() Mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup()	Chmod()Umask()Chown()

Run Levels

A run level is a state of init and the whole system that defines what system services are operating. Run levels are identified by numbers. Some system administrators use run

levels to define which subsystems are working, e.g., whether X is running, whether the network is operational, and so on.

- Whenever a LINUX system boots, firstly the **init** process is started which is actually responsible for running other start scripts which mainly involves initialization of you hardware, bringing up the network, starting the graphical interface.
- Now, the **init** first finds the default **runlevel** of the system so that it could run the start scripts corresponding to the default run level.
- A **runlevel** can simply be thought of as the state your system enters like if a system is in a single-user mode it will have a **runlevel 1** while if the system is in a multi-user mode it will have a **runlevel 5**.
- A **runlevel** in other words can be defined as a **preset single digit integer** for defining the operating state of your LINUX or UNIX-based operating system. Each runlevel designates a different system configuration and allows access to different combination of **processes**.

The important thing to note here is that there are differences in the runlevels according to the operating system. The standard **LINUX kernel** supports these seven different runlevels :

- 0 – System halt *i.e* the system can be safely powered off with no activity.
- 1 – Single user mode.
- 2 – Multiple user mode with no NFS(network file system).
- 3 – Multiple user mode under the command line interface (C L I)
- 4 – User-definable.
- 5 – Multiple user mode under GUI (graphical user interface) and this is the standard runlevel for most of the LINUX based systems.
- 6 – Reboot which is used to restart the system.