Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE(Autonomous)

Coimbatore – 49.

DEPARTMENT OF COMPUTER APPLICATIONS

COURSE : OPERATING SYSTEM

CLASS : I BCA 'B'

Unit I

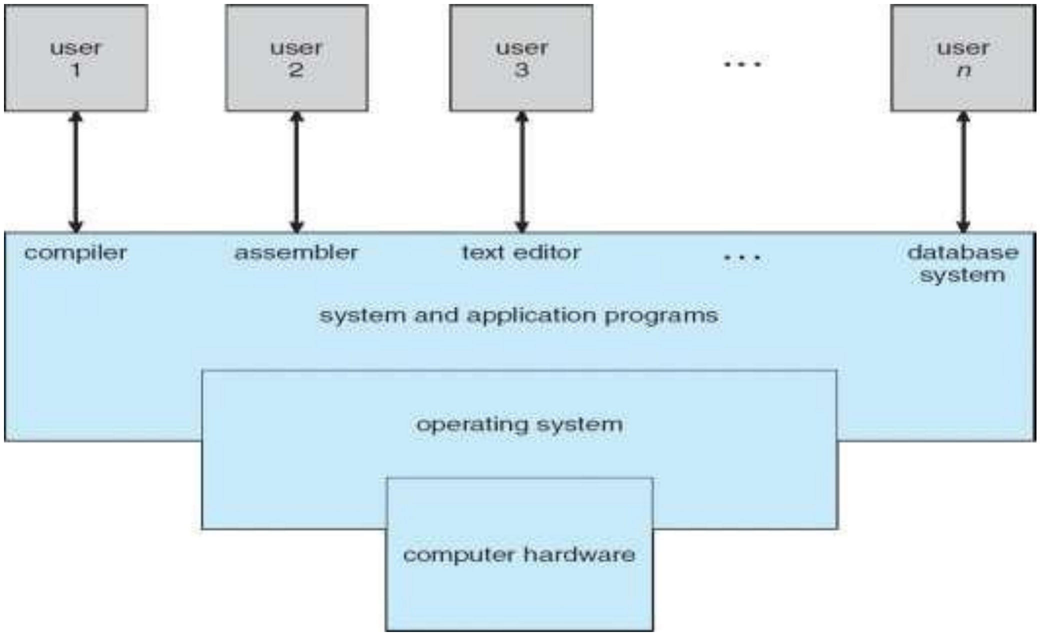
Introduction to OS

A program that acts as an intermediary between a user of a computer and thecomputer hardware

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

- Computer System Structure
- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers



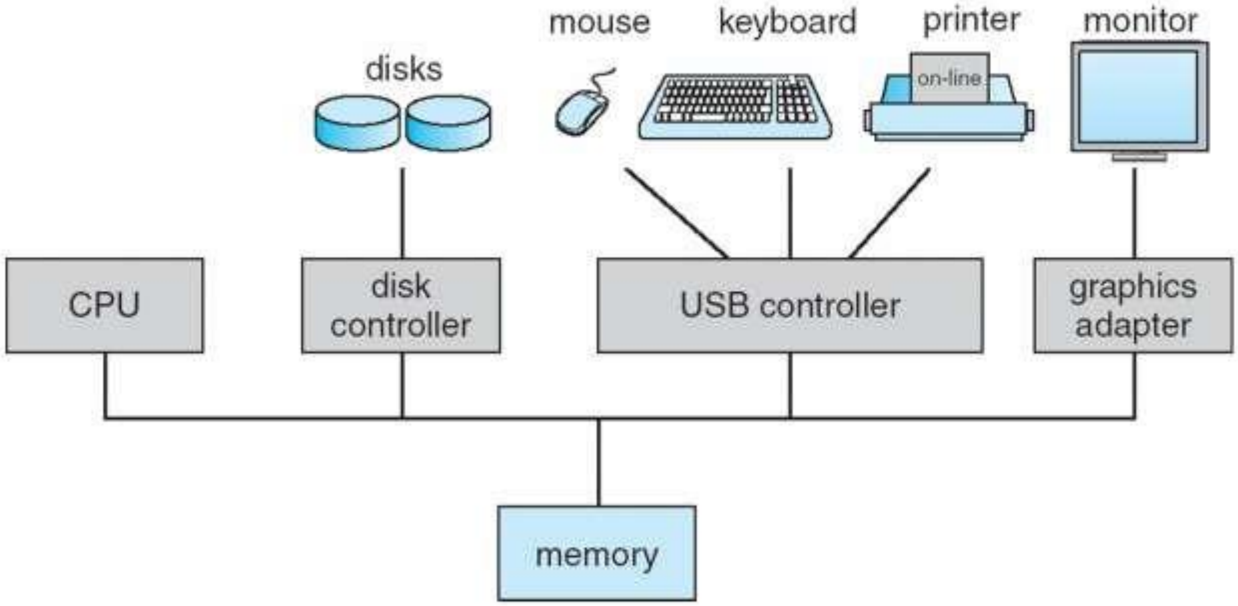
OS Definition

- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer

Computer Startup

- bootstrap program is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer System Organisation



- One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an *interrupt*
- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*

- A *trap* is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**
- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
- **polling**
- **vectored** interrupt system
- Separate segments of code determine what action should be taken for each typeof interrupt

I/O Structure

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion
  - **System call** – request to the operating system to allow user to wait for I/O completion
  - **Device-status table** contains entry for each I/O device indicating its type, address, and state
  - Operating system indexes into I/O device table to determine device statusand to modify table entry to include interrupt

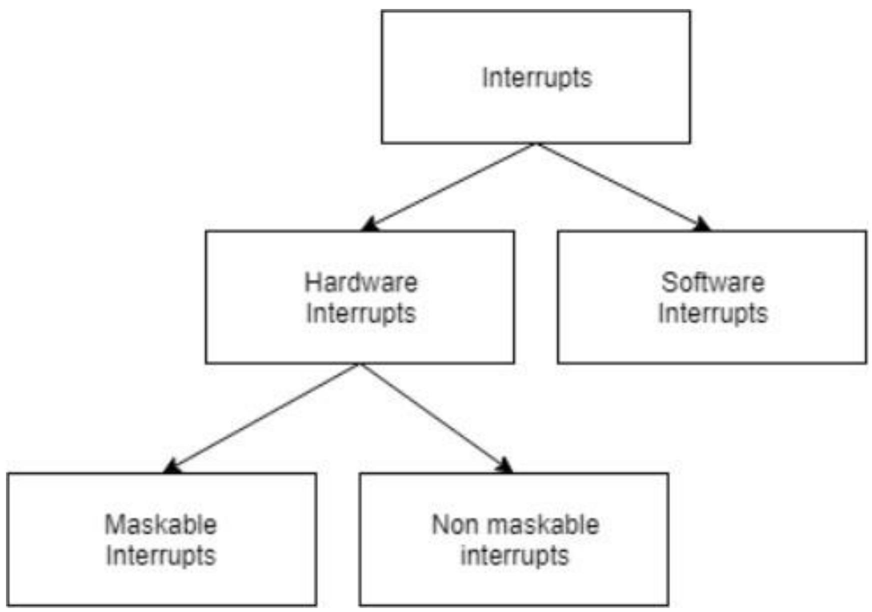
Storage Structure

- Main memory – only large storage media that the CPU can access directly
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity

- Magnetic disks – rigid metal or glass platters covered with magnetic recording material

Interrupt Handling:

- An interrupt is a necessary part of Computer System Organisation as it is triggered by hardware and software parts when they need immediate attention.
- An interrupt can be generated by a device or a program to inform the operating system to halt its current activities and focus on something else. The types of interrupts are better explained using the following diagram −



- Hardware and software interrupts are two types of interrupts. Hardware interrupts are triggered by hardware peripherals while software interrupts are triggered by software function calls.

- Hardware interrupts are of further two types. Maskable interrupts can be ignored or disabled by the CPU while this is not possible for non maskable interrupts.

Direct Memory Access Structure

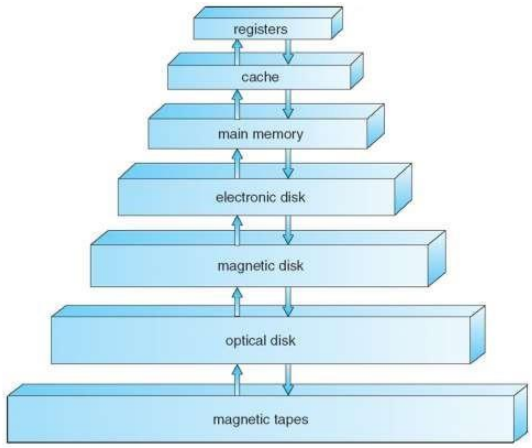
- Used for high-speed I/O devices able to transmit information at close to memory speeds

- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention

- Only one interrupt is generated per block, rather than the one interrupt perbyte

Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility

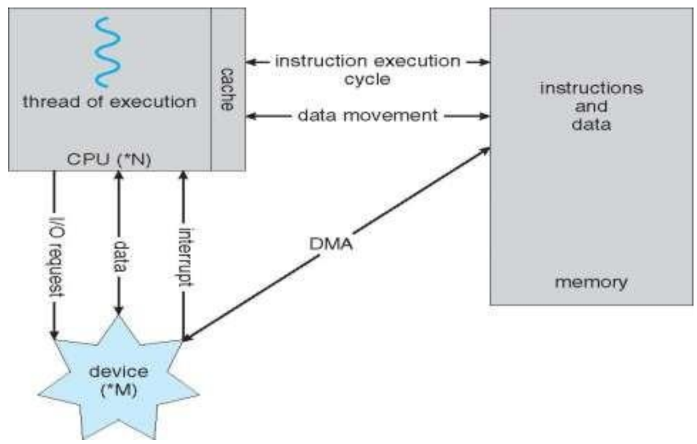


Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the deviceand the computer

Computer System Architecture

- Most systems use a single general-purpose processor (PDAs through mainframes)
    - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
    - Also known as parallel systems, tightly-coupled systems
    - Advantages include
        - Increased throughput
        - Economy of scale
        - Increased reliability – graceful degradation or fault tolerance
    - Two types
        - Asymmetric Multiprocessing
        - Symmetric Multiprocessing



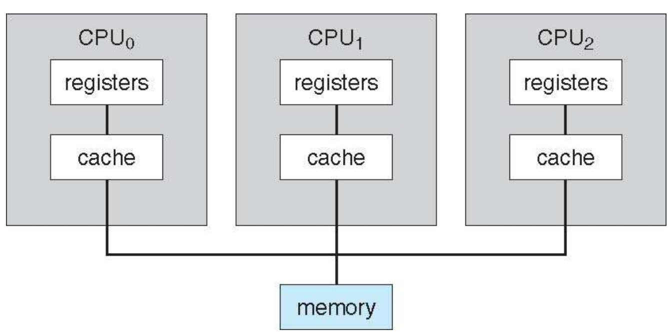


Fig: Symmetric multiprocessing architecture

OS Structure

n MS-DOS – written to provide the most functionality in the least space

  l Not divided into modules

  l Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated

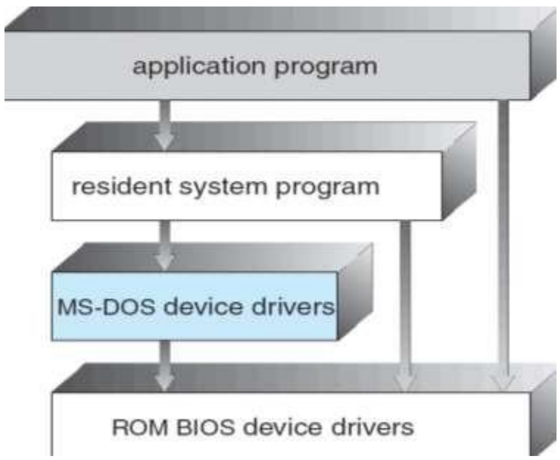


Fig: MS Dos structure

Layered Approach

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N)is the user interface.

- With modularity, layers are selected such that each uses functions (operations)and services of only lower-level layers

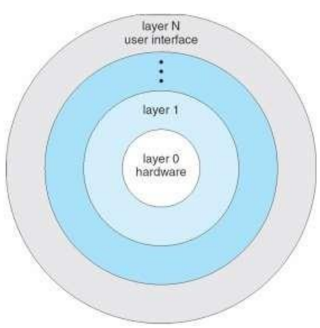


Fig: Layered System

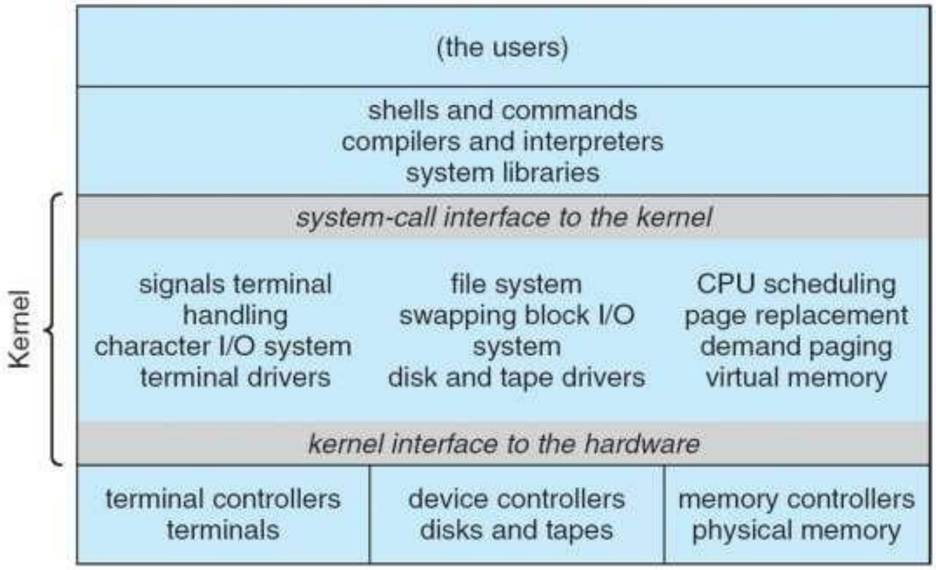


Fig: UNIX system structure

OPERATING SYSTEM STRUCTURE:

One of the most important aspects of operating systems is the ability to multiprogram. A single program cannot, in general, keep either the CPU or the I/O devices busy at all times. Single users frequently have multiple programs running. Multiprogramming increases CPUutilization by organizingjobs (code and data) so that the CPU always has one to execute.

The operating system keeps several jobs in memory simultaneously . Since, in general, main memory is too small to accommodate all jobs, the jobs are kept initially on the disk in the job pool.This pool consists of all processes residing on disk awaiting allocation of main memory.

Multiprogrammed systems provide an environment in which the various system resources (for example, CPU, memory, and peripheral devices) are utilized effectively, but they do not provide for user interaction with the computer system.

Time sharing requires an interactive computer system, which provides direct communication between the user and the system. The user gives instructions to the operating system or to a program directly, using a input device such as a keyboard, mouse, touch pad, or touch screen, and waits for immediate results on an output device. Accordingly, the response time shouldbe short—typically less than one second.

Time sharing and multiprogramming require that several jobs be kept simultaneously in memory. If several jobs are ready to be brought into memory,and if there is not enough room for all of them, then the system must choose among them.

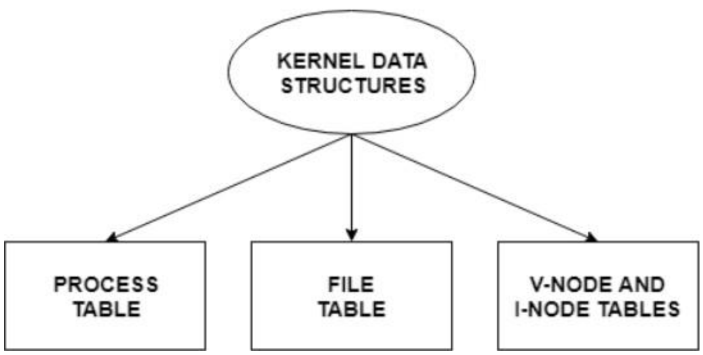
Kernel Data Structures:

The kernel data structures are very important as they store data about the current state of the system. For example, if a new process is created in the system, a kernel data structure is created that contains the details about the process.

Most of the kernel data structures are only accessible by the kernel and its subsystems. They may contain data as well as pointers to other data structures.

- ✓ Kernel Components

The kernel stores and organizes a lot of information. So it has data about which processes are running in the system, their memory requirements, files in use etc. To handle all this, three important structures are used. These are process table, file table and v node/ i node information.



Details about these are as follows:

- ✓ ***Process Table***

The process table stores information about all the processes running in the system. These include the storage information, execution status, file information etc.

When a process forks a child, its entry in the process table is duplicated including the file information and file pointers. So the parent and the child process share a file.

- ✓ ***File Table***

The file table contains entries about all the files in the system. If two or more processes use the same file, then they contain the same file information and the file descriptor number.

Each file table entry contains information about the file such as file status (file read or file write), file offset etc. The file offset specifies the position for next read or write into the file. The file table also contains v-node and i-node pointers which point to the virtual node and index node respectively. These nodes contain information on how to read a file.

✓ *V-Node and I-Node Tables*

Both the v-node and i-node are references to the storage system of the file and the storage mechanisms. They connect the hardware to the software.

The v-node is an abstract concept that defines the method to access file data without worrying about the actual structure of the system. The i-node specifies file access information like file storage device, read/write procedures etc.

System Call:

- In computing, a system call is the programmatic way in which a computer program requests a service from the kernel of the operatingsystem 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 arequest to the operating system's kernel.
- System call provides the services of the operating system to the userprograms via Application Program Interface(API). It provides an interface between a process and operating system to allow user-levelprocesses to request services of the operating system.
- System calls are the only entry points into the kernel system. Allprograms 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: end, abort, create, terminate, allocate and free memory.
2. File management: create, open, close, delete, read file etc.
3. Device management
4. Information maintenance
5. Communication

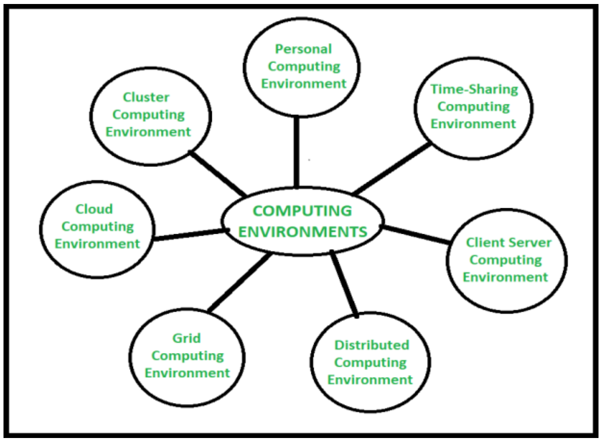
Computing Environments:

Computing Environments :

- When a problem is solved by the computer, during that computer uses many devices, arranged in different ways and which work together to solve problems.
- This constitutes a computing environment where various number of computer devices arranged in different ways to solve different types of problems in different ways.
- In different computing environments computer devices are arranged in different ways and they exchange information in between them to process and solve problem.
- One computing environment consists of many computers other computational devices, software and networks that to support processing and sharing information and solving task.
- Based on the organization of different computer devices and communication processes there exists multiple types of computing environments.

Types of Computing Environments :

There are the various types of computing environments. They are :



1. *Personal Computing Environment :*

   In personal computing environment there is a stand-alone machine. Complete program resides on computer and executed there. Different stand- alone machines that constitute a personal computing environment are laptops, mobiles, printers, computer systems, scanners etc. That we use at our homes and offices.

2. *Time-Sharing Computing Environment :*

In Time Sharing Computing Environment multiple users share system simultaneously. Different users (different processes) are allotted different time slice and processor switches rapidly among users according to it. For example, student listening to music while coding something in an IDE. Windows 95 and later versions, Unix, IOS, Linux operating systems are the examples of this time sharing computing environment.

3. *Client Server Computing Environment :*

   In client server computing environment two machines are involved i.e., client machine and server machine, sometime same machine also serve as client and server. In this computing environment client requests resource/service and server provides that respective resource/service. A server can provide service to multiple clients at a time and here mainly communication happens through computer network.

4. *Distributed Computing Environment :*

   In a distributed computing environment multiple nodes are connected together using network but physically they are separated. A single task is performed by different functional units of different nodes of distributed unit. Here different programs of an application run simultaneously on different nodes, and communication happens in between different nodes of this system over network to solve task.

5. *Grid Computing Environment :*

In grid computing environment, multiple computers from different locations works on single problem. In this system set of computer nodes running in cluster jointly perform a given task by applying resources of multiple computers/nodes. It is network of computing environment where several scattered resources provide running environment for single task.

6. *Cloud Computing Environment :*

In cloud computing environment on demand availability of computer system resources like processing and storage are availed. Here computing is not done in individual technology or computer rather it is computed in cloud of computers where all required resources are provided by cloud vendor. This environment primarily comprised of three services

i.e software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS).

7. *Cluster Computing Environment :*

In cluster computing environment cluster performs task where clusteris a set of loosely or tightly connected computers that work together. It is viewed as single system and performs task parallelly that's why also it is similar to parallel computing environment. Cluster aware applications are especially used in cluster computing environment.

OPEN SOURCE OPERATING SYSTEM:

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.

Linux lite:

Linux Lite is another free and open-source operating system that can run on lower-end hardware. It is a lightweight operating system designed to help users who are unfamiliar with Linux-based operating systems. The operating system includes all of the required programs, capabilities, tools, and desktops.

Chrome os:

Chrome OS is a partly open-source operating system with various attractive features. It's a part of the Chromium and Linux families, with features including better security, compatibility for supported Android and Chrome apps, Aura windows manager, Google cloud print, integrated media player, virtual desktop access, and cloud-based management. The only issue with the operating system is that it only supports Nexus devices or its hardware. As a result, if you're a Google fan, you'll love Chrome OS on a Chromebook.

Solaris:

Solaris is the commercial UNIX-based operating system of Sun Microsystems.Originally, Sun's **SunOS** operating system was based on BSD UNIX. Sunmoved to AT&T's System V UNIX as its base in 1991. In 2005, Sun open-sourced most of the Solaris code as the OpenSolaris project. The purchase of Sun by Oracle in 2009, however, left the state of this project unclear.

Advantages and Disadvantages of Open-Source Operating System

Various advantages and disadvantages of the open-source operating system are as follows:

Advantages:

1. *Reliable and efficient*

The open-source operating systems are most reliable and efficient. Thousandsof 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 arefar less expensive than commercially closed products.

3. *Flexibility*

xThe great advantage is you may customize it as per your requirement. Andthere 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 becausethe 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.