Operating system

An operating system, or "OS," is software that communicates with the hardware and allows other programs to run.

- It is comprised of system software, or the fundamental files your computer needs to boot up and function.
- Every desktop computer, tablet, and smartphone includes an operating system that provides basic functionality for the device.

System software

- is a type of software that manages and controls the hardware components of a computer and provides a platform for running application software. It serves as an intermediary between the hardware and user applications, ensuring that the computer's resources are used efficiently and effectively.

Open source os

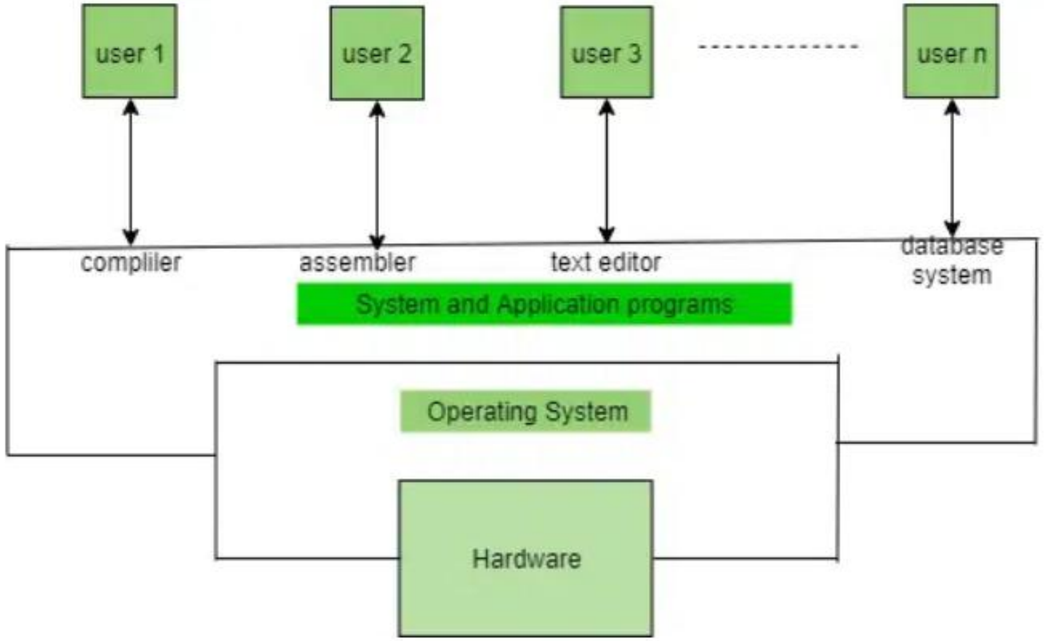
Open source refers to an openly distributed program code (including codes that are free of charge) that **can be utilized and modified by the end user without restriction.**

Android is an open-source operating system. The Android operating system is based on the Linux kernel, and its source code is open and freely available. This means that developers can access, modify, and distribute the Android source code under the terms of the open-source license

Closed source

- closed source software keeps its source code under wraps, accessible only to the creator or the authorized organization.

- **What is an operating system?**
- a. Software that manages hardware resources
- b. A set of application programs
- c. The physical components of a computer
- d. A type of antivirus software



Operating System

Functions of the Operating System

- **Resource Management:** The operating system manages and allocates memory, CPU time, and other hardware resources among the various programs and processes running on the computer.
- **Process Management:** The operating system is responsible for starting, stopping, and managing processes and programs. It also controls the scheduling of processes and allocates resources to them.
- **Memory Management:** The operating system manages the computer’s primary memory and provides mechanisms for optimizing memory usage.

- **Security:** The operating system provides a secure environment for the user, applications, and data by implementing security policies and mechanisms such as access controls and encryption.
- **Job Accounting:** It keeps track of time and resources used by various jobs or users.
- **File Management:** The operating system is responsible for organizing and managing the file system, including the creation, deletion, and manipulation of files and directories.
- **Device Management:** The operating system manages input/output devices such as printers, keyboards, mice, and displays. It provides the necessary drivers and interfaces to enable communication between the devices and the computer.

- **Networking:** The operating system provides networking capabilities such as establishing and managing network connections, handling network protocols, and sharing resources such as printers and files over a network.
- **User Interface**: The operating system provides a user interface that enables users to interact with the computer system. This can be a Graphical User Interface (GUI), a Command-Line Interface (CLI), or a combination of both.

- **1.** Which of the following is *not* a core function of an Operating System?
  A) Process management
  B) Memory management
  C) Compiling source code
  D) File management

2. The part of the OS that loads the operating system into memory when the system is powered on is called:
A) Kernel
B) Bootloader
C) Scheduler
D) BIOS

Modes of OS

- In an operating system (OS), **supervisor mode** (often referred to as **kernel mode**) and **user mode** are two distinct modes of operation for the CPU. These modes control the level of access that running code has to system resources, ensuring both security and stability.

Kernel Mode (Supervisor Mode)

- Full Access: In kernel mode, the CPU has unrestricted access to all system resources, including hardware (like memory, I/O devices) and system data. Code running in kernel mode can execute any CPU instruction and reference any memory address.
- Critical Functions: This mode is used to execute critical parts of the operating system, such as process management, memory management, and hardware communication. Because the OS kernel operates in this mode, it can directly manage hardware and perform tasks that require high privileges.

- **3.** Which of the following best describes the function of a **kernel**?
  A) Provides user interface for system interaction
  B) Manages system resources and acts as a bridge between hardware and software
  C) Handles application execution only
  D) Monitors network traffic
- **4.** The main objective of **multiprogramming** is:
  A) To increase CPU utilization
  B) To reduce memory usage
  C) To increase I/O speed
  D) To increase file access time

- **Example**: When you save a file, the OS must write data to the disk, manage the file system, and interact with the storage hardware. These operations require kernel mode because they involve direct hardware access and modifications to system data structures.

User Mode

- **Restricted Access**: In user mode, the CPU has limited access to system resources. Code running in user mode cannot directly interact with hardware or reference protected memory. If a program in user mode attempts to perform a restricted operation (like directly accessing hardware), it will result in an error or exception.

- **Example**: A web browser running in user mode can request the OS to open a file or network connection. However, it does so through system calls to the OS, which performs these operations in kernel mode on behalf of the browser. If the browser were allowed to operate in kernel mode, any bug could potentially crash the system or corrupt data.

Q1

- **Which of the following is an example of an Operating System?**
- a) Microsoft Word
- b) Adobe Photoshop
- c) Linux
- d) Google Chrome

Q2

- **Which of the following is NOT a function of an Operating System?**
- a) Memory management
- b) Process management
- c) Data management
- d) Compilation of code

Q3

- **Which part of the operating system is responsible for managing files?**
- a) Kernel
- b) File System
- c) Shell
- d) Process Scheduler

- **1.** Which of the following operations **requires the operating system to run in kernel mode**?
  A) Reading a variable from user memory
  B) Switching from one process to another
  C) Performing arithmetic operations
  D) Accessing data in user space

OS Evolution

- **Early Computers (1940s-1950s): Pre-OS Era**
- **Manual Operation**: Early computers like the ENIAC (Electronic Numerical Integrator and Computer) and the UNIVAC I (Universal Automatic Computer) did not have operating systems. Users interacted with these machines directly, using switches, plugs, and punched cards to input programs and data.
- **Program Loading**: Every program had to be manually loaded into the computer, and the process was labor-intensive and error-prone. There was no automation for managing tasks or resources.

Batch Processing Systems (1950s-1960s): First Generation OS

- Batch Processing: To improve efficiency, batch processing systems were introduced. Programs and jobs were collected into batches and executed sequentially without user intervention during execution.
- Early Operating Systems:IBM GM-NAA I/O (1956): One of the first OSs for managing I/O operations and job scheduling.
- IBM's OS/360 (1964): A major advancement in multiprogramming and batch processing, capable of handling multiple jobs concurrently and managing resources efficiently.

Multi-Programming and Time-Sharing (1960s): Second Generation OS

- Multi-Programming: Enabled the computer to run multiple programs simultaneously by sharing CPU time among them. This improved system utilization and efficiency.
- Time-Sharing Systems: Allowed multiple users to interact with the computer concurrently. The OS managed each user’s session and ensured fair access to resources.

- Compatible Time-Sharing System (CTSS): Developed at MIT, CTSS allowed multiple users to run programs and access data simultaneously, marking the beginning of modern timesharing systems.
- Multics: A pioneering system developed at MIT, which introduced many features that influenced future OS design, including hierarchical file systems and user accounts.

- **2.** The **Supervisor (Kernel) mode** is primarily used for:
  A) Executing user applications
  B) Restricting access to system calls
  C) Performing privileged system operations
  D) Handling arithmetic operations only

- **5.** The **main reason for having two modes (user and kernel)** in an operating system is:
  A) To simplify user interface design
  B) To separate user programs from system programs
  C) To protect the OS and critical resources from user programs
  D) To speed up the CPU

Unix and Portable Operating Systems (1970s): Third Generation OS

- Unix: Developed at AT&T’s Bell Labs by Ken Thompson, Dennis Ritchie, and others, Unix introduced several innovative concepts:Portability:
- Written in the C programming language, Unix could be easily modified to run on different hardware.
- Modular Design: Unix used small, single-purpose programs that could be combined to perform complex tasks.
- File System: A hierarchical file system allowed for organized storage and retrieval of data.
- Process Management: Included support for multitasking, process control, and inter-process communication.
- Impact: Unix's design influenced many other operating systems, including Linux and BSD (Berkeley Software Distribution).

Personal Computers and Graphical User Interfaces (1980s): Fourth Generation OS

- MS-DOS: Microsoft’s Disk Operating System was designed for IBM PCs. It was command-line based, requiring users to type commands to manage files and run applications.
- Graphical User Interfaces (GUIs): The introduction of GUIs revolutionized user interaction with computers:
- Apple Macintosh (1984): Introduced a user-friendly GUI with windows, icons, and menus, making computers accessible to non-technical users.
- Microsoft Windows: Initially launched as a GUI shell for MS-DOS, Windows evolved into a full operating system with Windows 95, incorporating advanced features like multitasking, plug-and-play hardware support, and networking capabilities.

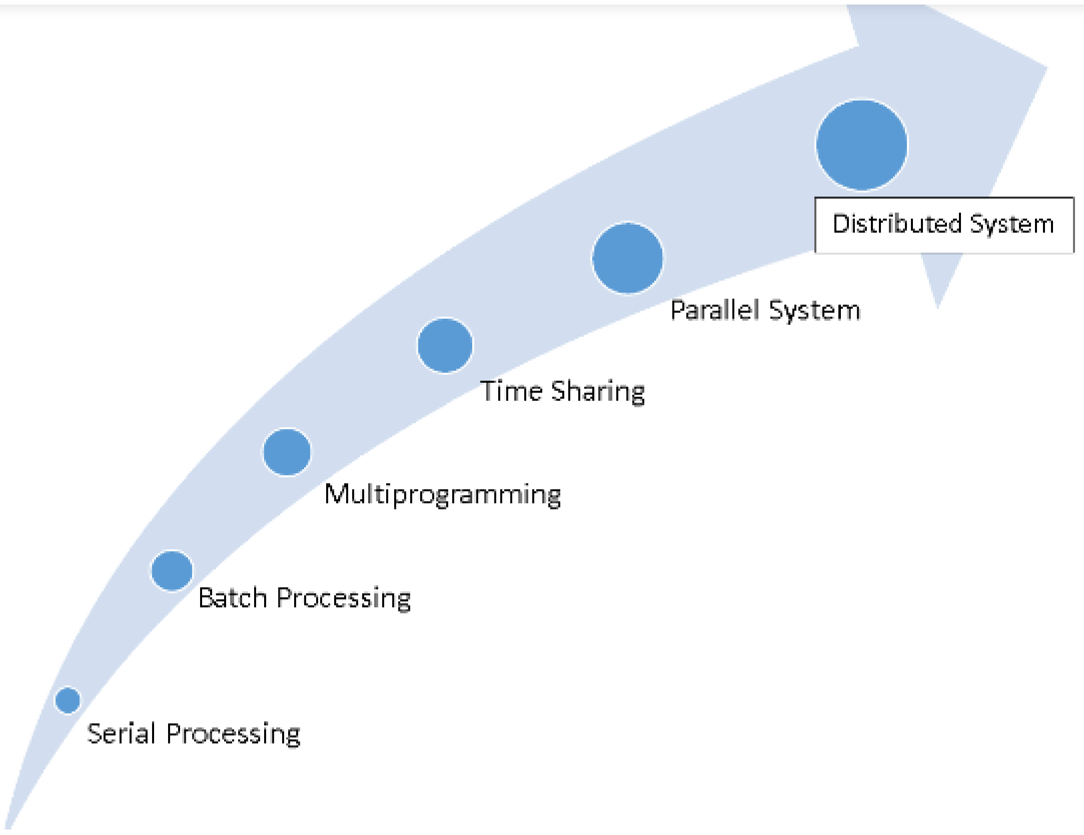
Networking and Distributed Systems (1990s): Fifth Generation OS

- Networking Integration: Operating systems began integrating networking capabilities, allowing computers to connect to networks and communicate with other systems.
- Distributed Systems: The OS had to manage distributed resources and coordinate tasks across multiple machines.
- Network File Systems: Systems like NFS (Network File System) allowed files to be accessed across a network as if they were local.
- Cluster Computing: Operating systems managed clusters of computers working together to perform tasks, improving performance and fault tolerance.

Modern Operating Systems (2000s-Present): Sixth Generation OS

- **Windows, macOS, and Linux**:**Windows**: Continued to evolve with versions like Windows XP, Windows 7, and Windows 10, focusing on user interface improvements, security features, and support for a wide range of applications and hardware.
- **macOS**: Apple's OS for Mac computers, known for its sleek design, integrated applications, and UNIX-based foundation.
- **Linux**: An open-source OS that has gained popularity for servers, desktops, and embedded systems due to its flexibility, security, and strong community support

Evolution of os



Serial Processing

- If a programmer wishes to execute a program on those days, the following serial steps are necessary.
- Type the program or punched card.
- Convert the punched card to a card reader.
- submit to the computing machine, is there any errors, the error was indicated by the lights.
- The programmer examined the register and main memory to identify the cause of an error
- Take outputs on the printers.
- Then the programmer ready for the next program.

Drawback

- This type of processing is difficult for users, it takes much time and the next program should wait for the completion of the previous one. The programs are submitted to the machine one after one, therefore the method is said to be **serial processing**.

Batch processing

Batch processing system means to grab all types of programs and data in the batch form then proceed to process. Main motive of using batch processing system is to decrease the set up time while submitting the similar jobs to CPU.

jobs are divided into groups, and finally precede same jobs to similar batch. Now all batched jobs are ready to execution one by one without wasting more time, and due to this system enhance the system utilization while decrease the turnaround time. Example pyroll

Multiprogramming

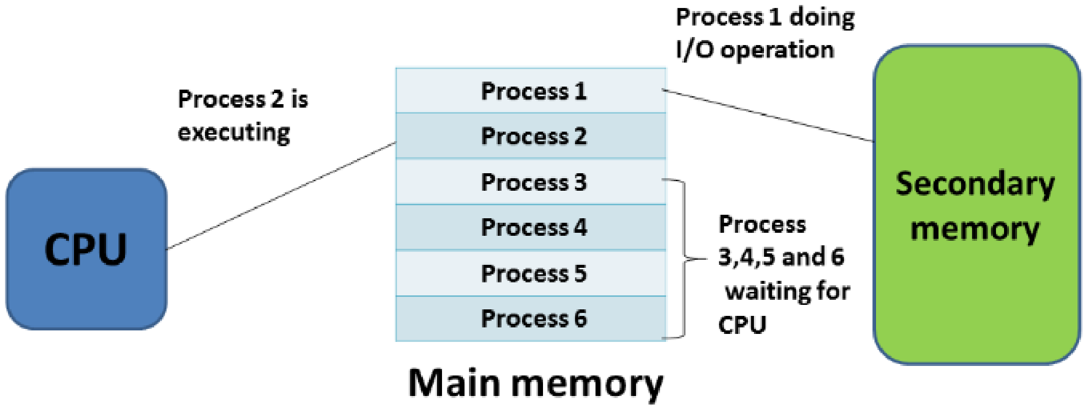
- Multiprogramming is a technique where multiple programs are loaded into the computer's memory simultaneously.
- The CPU is switched rapidly between these programs, giving the illusion that they are executing concurrently.
- The primary goal of multiprogramming is to keep the CPU busy and improve overall system efficiency by allowing it to work on another task while one task is waiting for I/O (Input/Output) operations, such as reading from or writing to a disk

- However, because there is only one processor, there is no true simultaneous execution of different programs. Instead, the operating system (OS) executes part of one program, then part of another, and so on. In this sense, multiprogramming can be thought of as pseudo-parallelism. To the user, it appears that multiple programs are executing at the same time, but that is not what is happening.

- Multiprogramming addresses this issue by allowing multiple programs to load into memory and run each one in rotation as CPU resources become available. For example, when Program A starts, the operating system assigns CPU resources to that program until the program launches into its I/O operations. Then, the OS assigns CPU resources to Program B, which is already loaded into memory. If Program B launches into I/O operations and Program A is still running its I/O operations, the OS will assign CPU resources to Program C; otherwise, it will assign them back to Program A.

- Prior to the introduction of multiprogramming, single processor computers could run only one program at a time. Once the program was launched, it ran to completion, unless it was interrupted or the application ceded control. Only after the first program finished running could the processor execute the next program in the queue. This meant that the CPU sat idle during I/O operations, even if other programs were waiting, resulting in application delays and underutilized processor resources.

- There are different programs that want to get executed. So these programs are kept in the ready queue. And are assigned to the CPU one by one. If one process gets blocked then other processes from the ready queue are assigned to the CPU. The aim of this is optimal resource utilization and more CPU utilization. In the below figure, different processes are there in RAM(main memory). Some processes are waiting for the CPU, and process 2(which was previously executing) is now doing I/O operations. So CPU shifted to execute process 1.



CPU

Process 2 is executing

Process 1

Process 2

Process 3

Process 4

Process 5

Process 6

Main memory

Process 1 doing I/O operation

Process 3,4,5 and 6 waiting for CPU

Secondary memory

- **Examples of Multiprogramming Operating Systems**
- Following are the multiprogramming operating system example:
- Desktop operating systems, including Windows, macOS, and various Linux distributions. These are contemporary operating systems that make use of a variety of multiprogramming concepts. A system running one of these (or more) operating systems allows a user to run multiple jobs at once. For instance, many games have been developed to utilize just one processor core.
- One can send and receive text messages while simultaneously listening to music on a phone running Android, iOS, or another mobile operating system.
- application software, including media players, Office, and well-known web browsers. Any modern web browser would allow a user to open as many windows or tabs as necessary in order to visit multiple websites at once. You can enroll in the **best Full Stack Development course online** to create an excellent web application.

- CPU utilization is high because the CPU is never goes to idle state.
- Memory utilization is efficient.
- CPU throughput is high and also supports multiple interactive user terminals.

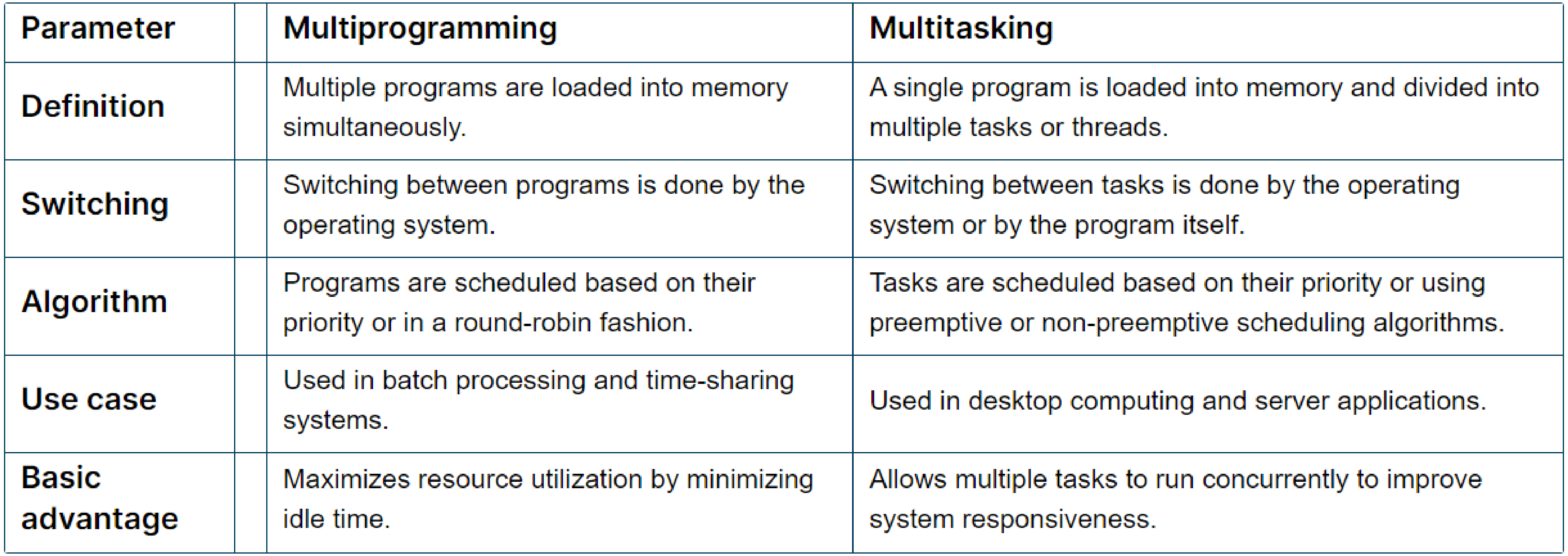
- Multiprogramming and Multitasking are two terms commonly used in computer science to describe different approaches to running multiple programs or tasks on a computer system. These terms are often used interchangeably. They have distinct meanings and use cases

- Multiprogramming refers to a technique used in **operating systems** to maximize the use of computer resources. The basic idea behind multiprogramming is to allow multiple programs to run on a single processor simultaneously by partitioning the processor’s time and memory resources between them.
- In a multiprogramming system, the operating system loads several programs into memory simultaneously and then switches between them in a way that gives the illusion of simultaneous execution. This technique allows the computer system to make the most efficient use of its resources by minimizing idle time and maximizing the throughput of multiple programs

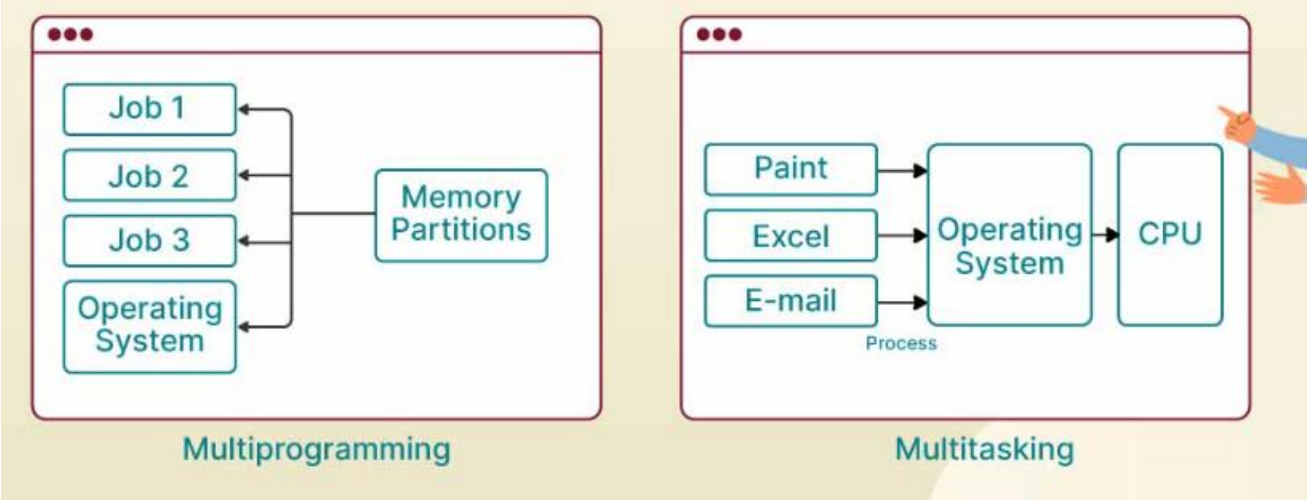
- On Android, iOS, and other mobile operating systems on the phone, one can search on Google while sending and receiving text messages on what’s the app at one time. This means these two programs are in the memory, and the CPU is switching between them.

Multitasking

- Multitasking refers to the ability of an operating system to run multiple tasks or processes concurrently on a computer system. **Multitasking operating system** uses time-sharing to switch between tasks or processes in a way that gives the illusion of parallel execution. Unlike multiprogramming, where multiple programs are loaded into memory simultaneously, in Multitasking, a single program is loaded into memory and divided into multiple tasks or threads that can be executed concurrently



Parameter		Multiprogramming	Multitasking
Definition		Multiple programs are loaded into memory simultaneously.	A single program is loaded into memory and divided into multiple tasks or threads.
Switching		Switching between programs is done by the operating system.	Switching between tasks is done by the operating system or by the program itself.
Algorithm		Programs are scheduled based on their priority or in a round-robin fashion.	Tasks are scheduled based on their priority or using preemptive or non-preemptive scheduling algorithms.
Use case		Used in batch processing and time-sharing systems.	Used in desktop computing and server applications.
Basic advantage		Maximizes resource utilization by minimizing idle time.	Allows multiple tasks to run concurrently to improve system responsiveness.



Multiprogramming

Multitasking

Parallel System

- There is a trend multiprocessor system, such system have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices.
- These systems are referred to as "Tightly Coupled" system. Then the system is called **a parallel system**. In the parallel system, a number of processors are executing there job in parallel.

- Parallel operating systems are designed to speed up the execution of programs by dividing them into multiple segments.
- It is used for dealing with **multiple processors simultaneously** by using computer resources which include a single computer with multiple processors and several computers connected by a network to form a cluster of parallel processing or a combination of both.

Distributed

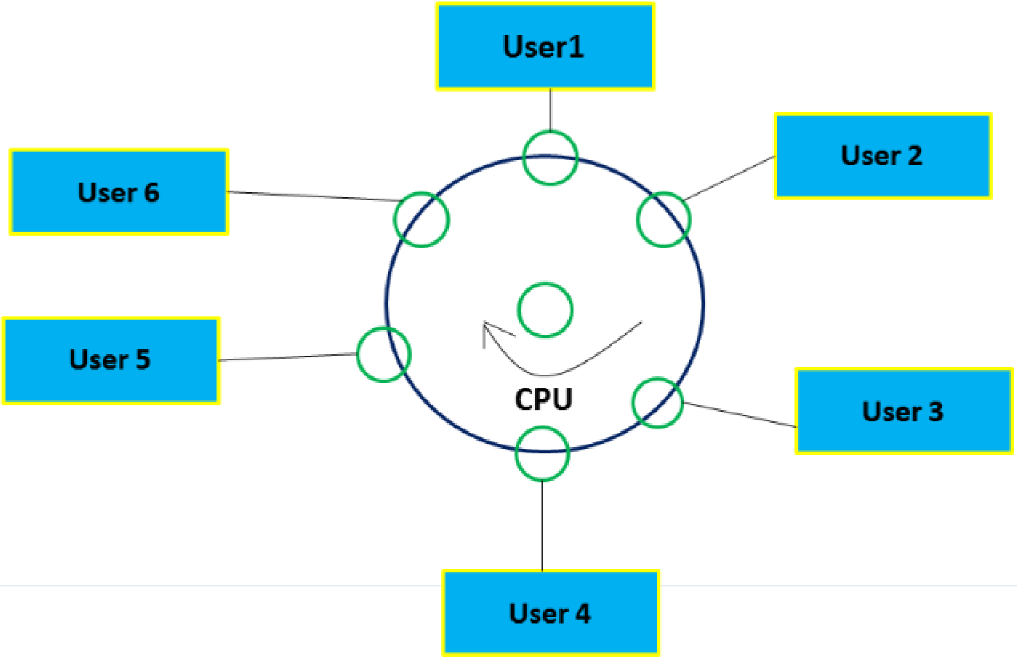
- In a **distributed operating system**, the processors cannot share a memory or a clock, each processor has its own local memory. The processor communicates with one another through various communication lines, such as high-speed buses. These systems are referred to as "Loosely Coupled" systems.

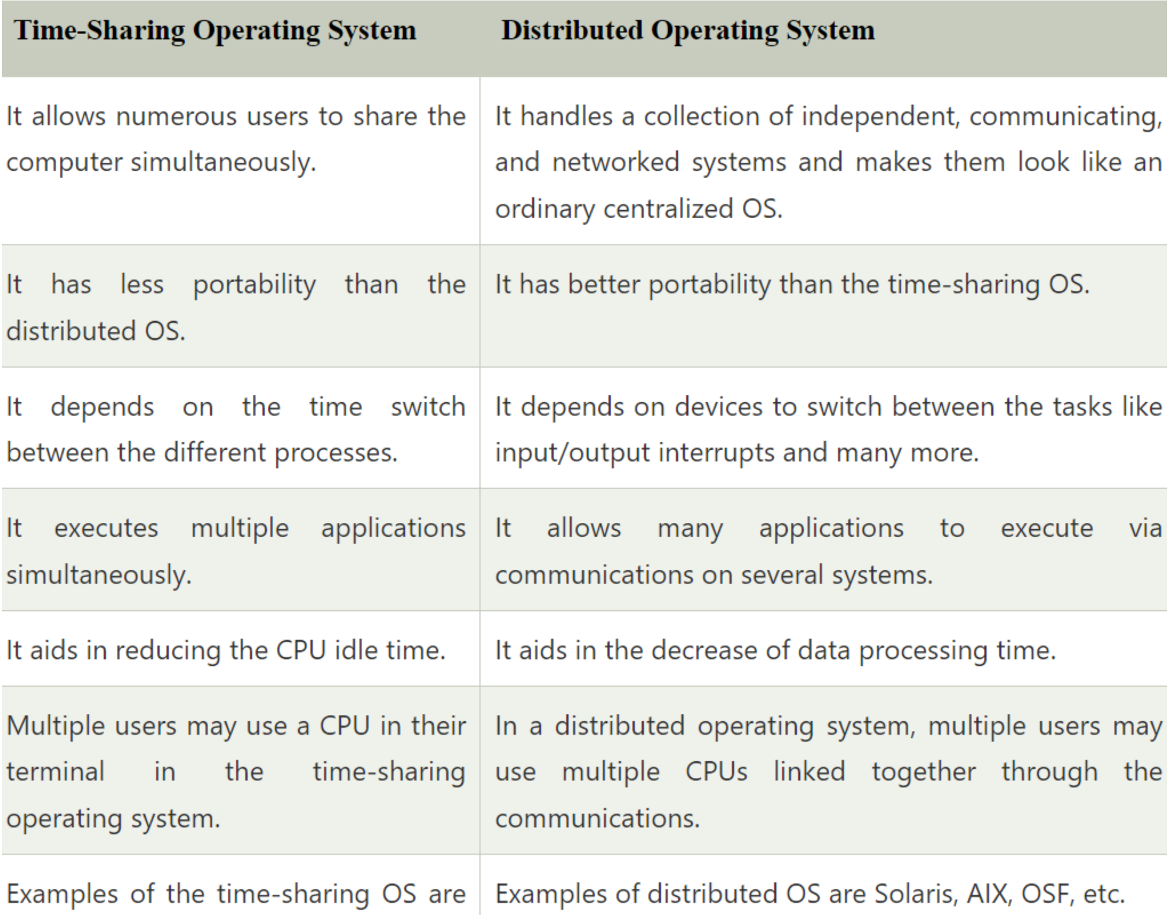
- Distributed systems are usually used to distribute software applications and data. Distributed systems are also used to manage the resources of multiple computers. Users could be at different sites. Multiple computers are connected via a single communication channel. Every system has its own processor and memory.
- Resources like disk, computer, CPU, network interface, nodes, etc., are shared among different computers at different locations. It increases data availability in the entire system.

- **Advantages**
- It is more reliable as a failure of one system will not impact the other computers or the overall system.
- All computers work independently.
- The system works at a higher speed as resources are shared
- The host system has less load.
- Computers can be easily added to the system.
- **Disadvantages**
- Costly setup.
- If the server fails, then the whole system will fail.
- Complex software is used for such a system

Time-sharing OS

- A time sharing operating system is a type of computer operating system that allows multiple users to interact with a single computer system at the same time. The OS achieves this by dividing the CPU time, memory, and other resources of the computer among different users, allowing each user to perform their tasks independently as if they were the sole user of the system.
- UNIX
- Linux


User1
User 2
User 6
User 5
CPU
User 3
User 4



Time-Sharing Operating System	Distributed Operating System
It allows numerous users to share the computer simultaneously.	It handles a collection of independent, communicating, and networked systems and makes them look like an ordinary centralized OS.
It has less portability than the distributed OS.	It has better portability than the time-sharing OS.
It depends on the time switch between the different processes.	It depends on devices to switch between the tasks like input/output interrupts and many more.
It executes multiple applications simultaneously.	It allows many applications to execute via communications on several systems.
It aids in reducing the CPU idle time.	It aids in the decrease of data processing time.
Multiple users may use a CPU in their terminal in the time-sharing operating system.	In a distributed operating system, multiple users may use multiple CPUs linked together through the communications.
Examples of the time-sharing OS are	Examples of distributed OS are Solaris, AIX, OSF, etc.