What is an Operating System?

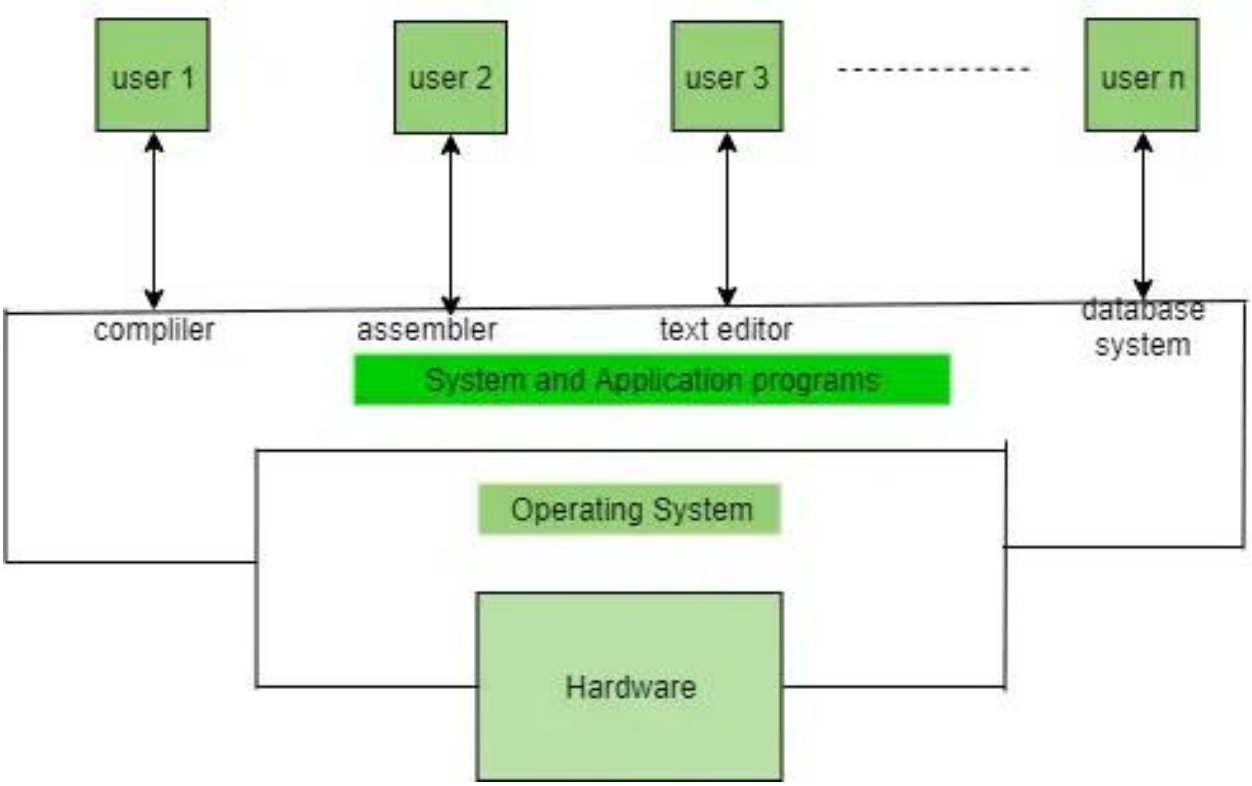
Operating System lies in the category of system software. It basically manages all the resources of the computer. An operating system acts as an interface between the software and different parts of the computer or the computer hardware. The operating system is designed in such a way that it can manage the overall resources and operations of the computer.

Operating System is a fully integrated set of specialized programs that handle all the operations of the computer. It controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer. Examples of Operating Systems are Windows, Linux, Mac OS, etc.

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is the most important type of system software in a computer system.

What is an Operating System Used for?

The operating system helps in improving the computer software as well as hardware. Without OS, it became very difficult for any application to be user-friendly. The Operating System provides a user with an interface that makes any application attractive and user-friendly. The operating System comes with a large number of device drivers that make OS services reachable to the hardware environment. Each and every application present in the system requires the Operating System. The operating system works as a communication channel between system hardware and system software. The operating system helps an application with the hardware part without knowing about the actual hardware configuration. It is one of the most important parts of the system and hence it is present in every device, whether large or small device.



Operating System

For more, refer to Need of Operating Systems.

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.
- **Backup and Recovery:** The operating system provides mechanisms for backing up data and recovering it in case of system failures, errors, or disasters.
- **Virtualization:** The operating system provides virtualization capabilities that allow multiple operating systems or applications to run on a single physical machine. This can enable efficient use of resources and flexibility in managing workloads.
- **Performance Monitoring:** The operating system provides tools for monitoring and optimizing system performance, including identifying bottlenecks, optimizing resource usage, and analyzing system logs and metrics.
- **Time-Sharing:** The operating system enables multiple users to share a computer system and its resources simultaneously by providing time-sharing mechanisms that allocate resources fairly and efficiently.
- **System Calls:** The operating system provides a set of system calls that enable applications to interact with the operating system and access its resources. System calls provide a standardized interface between applications and the operating system, enabling portability and compatibility across different hardware and software platforms.

- **Error-detecting Aids:** These contain methods that include the production of dumps, traces, error messages, and other debugging and error-detecting methods.

For more, refer to Functions of Operating System.

Objectives of Operating Systems

Let us now see some of the objectives of the operating system, which are mentioned below.

- **Convenient to use:** One of the objectives is to make the computer system more convenient to use in an efficient manner.
- **User Friendly:** To make the computer system more interactive with a more convenient interface for the users.
- **Easy Access:** To provide easy access to users for using resources by acting as an intermediary between the hardware and its users**.**
- **Management of Resources:** For managing the resources of a computer in a better and faster way.
- **Controls and Monitoring:** By keeping track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- **Fair Sharing of Resources:** Providing efficient and fair sharing of resources between the users and programs.

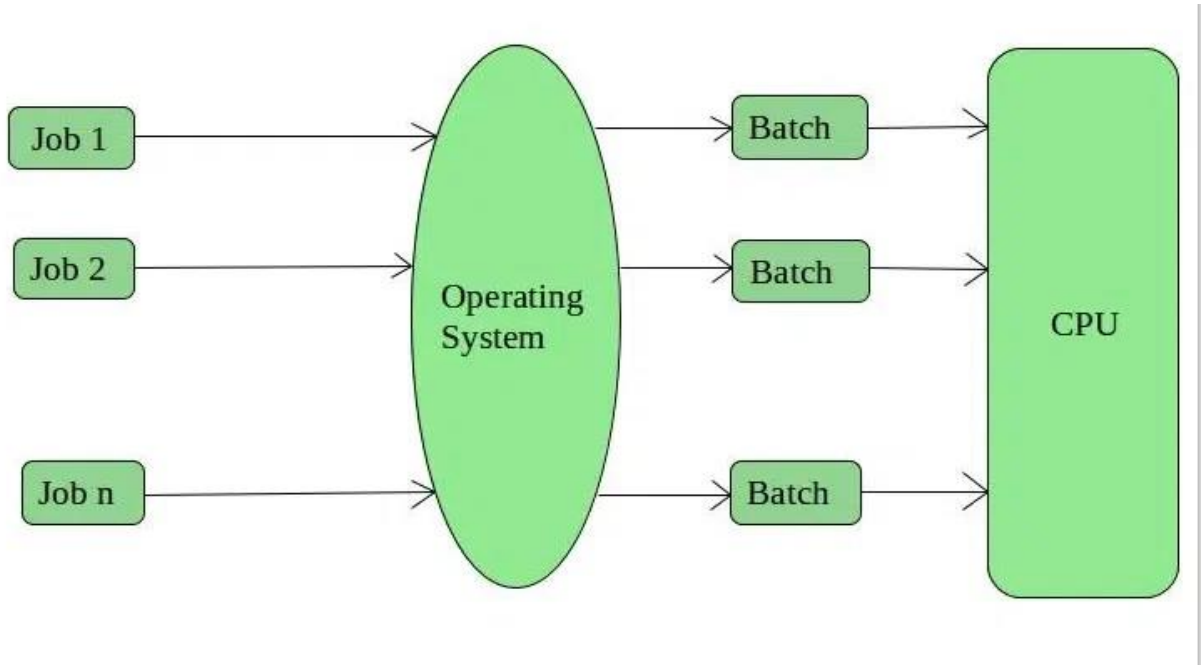
Types of Operating Systems

There are several types of Operating Systems which are mentioned below.

- Batch Operating System
- Multi-Programming System
- Multi-Processing System
- Multi-Tasking Operating System
- Time-Sharing Operating System
- Distributed Operating System
- Network Operating System
- Real-Time Operating System
- Multithreading Operating System

1. Batch Operating System

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirements and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs. Batch Operating System is designed to manage and execute a large number of jobs efficiently by processing them in groups.



Advantages of Batch Operating System

- Multiple users can share the batch systems.
- The idle time for the batch system is very less.
- It is easy to manage large work repeatedly in batch systems.

Disadvantages of Batch Operating System

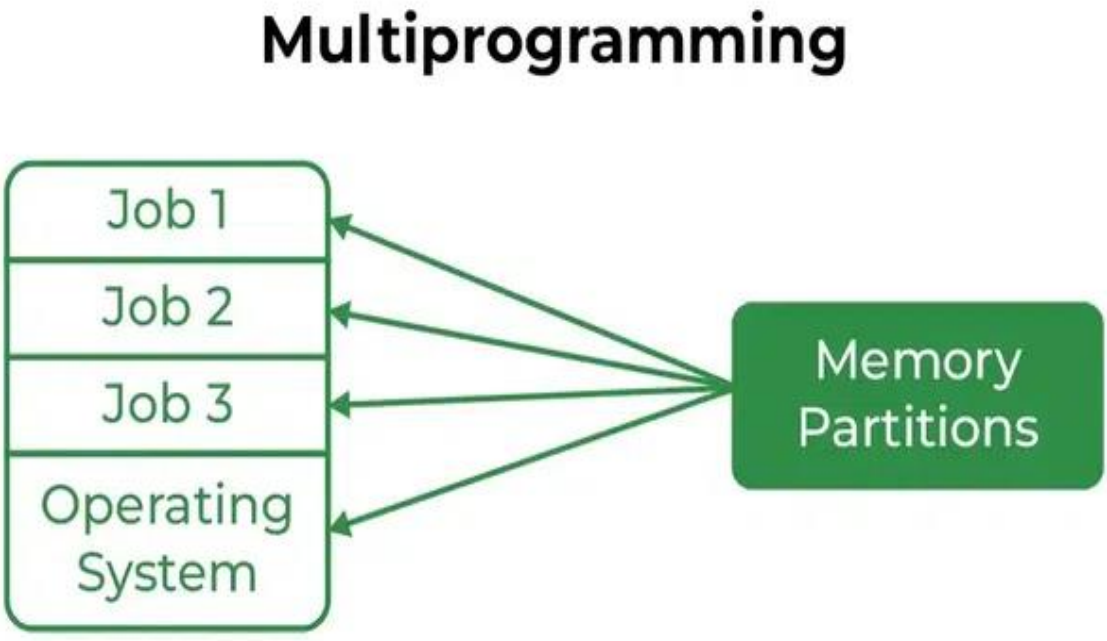
- Batch systems are hard to debug.
- It is sometimes costly.
- The other jobs will have to wait for an unknown time if any job fails.

- In batch operating system the processing time for jobs is commonly difficult to accurately predict while they are in the queue.
- It is difficult to accurately predict the exact time required for a job to complete while it is in the queue.

Examples of Batch Operating Systems: Payroll Systems, Bank Statements, etc.

2. Multi-Programming Operating System

Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution. This is basically used for better execution of resources.



Advantages of Multi-Programming Operating System

- Multi Programming increases the Throughput of the System.

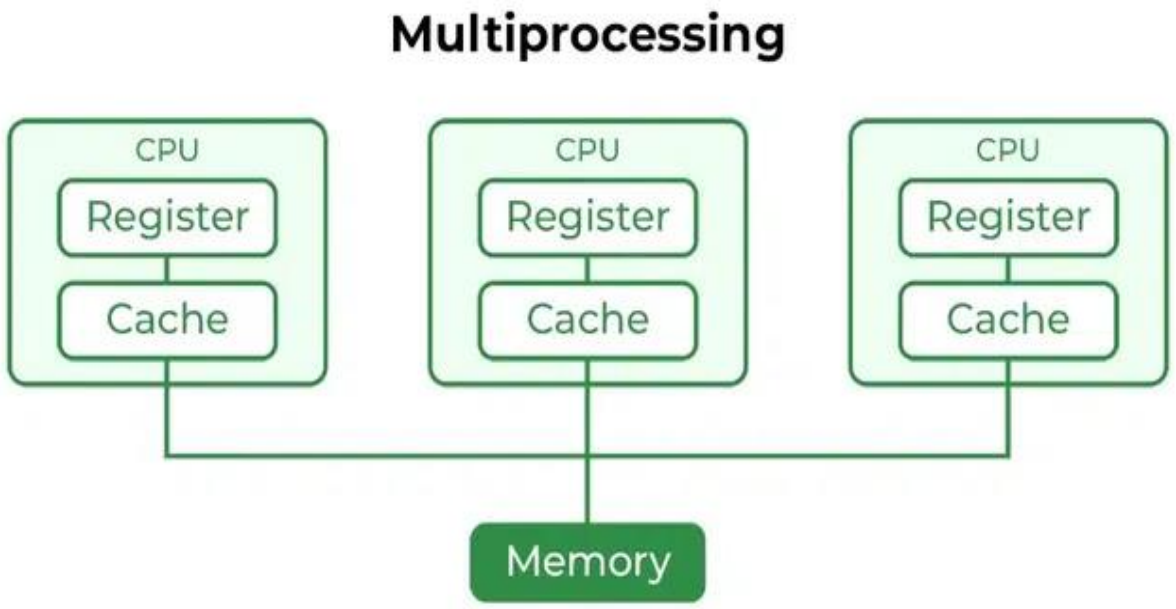
- It helps in reducing the response time.

Disadvantages of Multi-Programming Operating System

- There is not any facility for user interaction of system resources with the system.

3. Multi-Processing Operating System

Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources. It betters the throughput of the System.



Advantages of Multi-Processing Operating System

- It increases the throughput of the system.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

Disadvantages of Multi-Processing Operating System

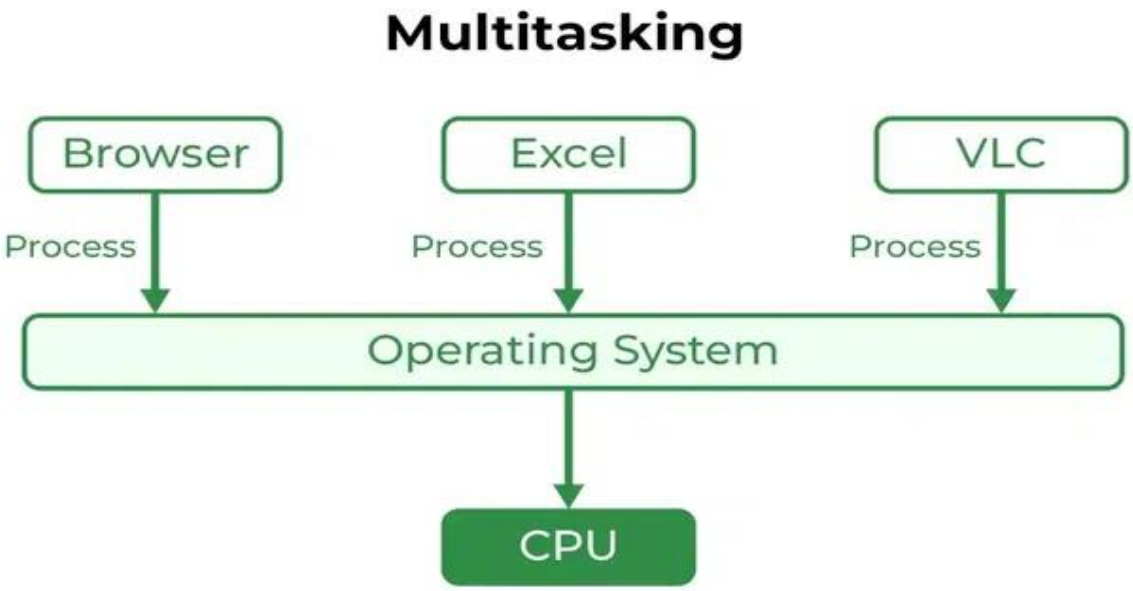
- Due to the multiple CPU, it can be more complex and somehow difficult to understand.

4. Multi-Tasking Operating System

Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling Algorithm. It can run multiple programs simultaneously.

There are two types of Multi-Tasking Systems which are listed below.

- Preemptive Multi-Tasking
- Cooperative Multi-Tasking



Advantages of Multi-Tasking Operating System

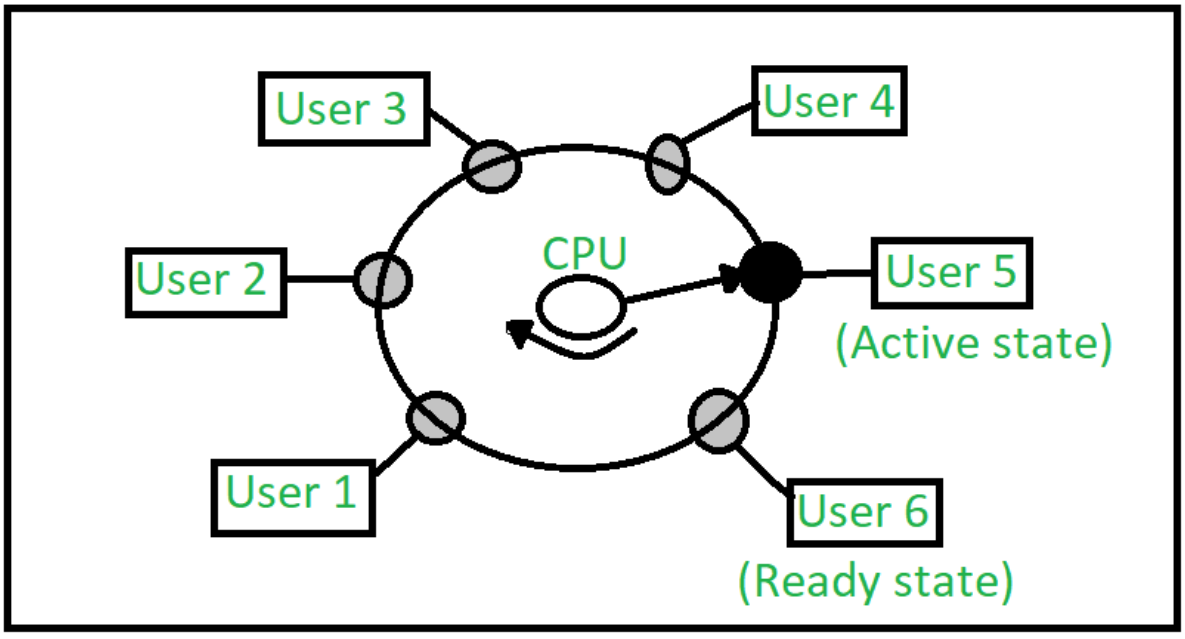
- Multiple Programs can be executed simultaneously in Multi-Tasking Operating System.
- It comes with proper memory management.

Disadvantages of Multi-Tasking Operating System

- The system gets heated in case of heavy programs multiple times.

5. Time-Sharing Operating Systems

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of the CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



Time-Sharing OS

Advantages of Time-Sharing OS

- Each task gets an equal opportunity.
- Fewer chances of duplication of software.
- CPU idle time can be reduced.
- Resource Sharing: Time-sharing systems allow multiple users to share hardware resources such as the CPU, memory, and peripherals, reducing the cost of hardware and increasing efficiency.

- Improved Productivity: Time-sharing allows users to work concurrently, thereby reducing the waiting time for their turn to use the computer. This increased productivity translates to more work getting done in less time.
- Improved User Experience: Time-sharing provides an interactive environment that allows users to communicate with the computer in real time, providing a better user experience than batch processing.

Disadvantages of Time-Sharing OS

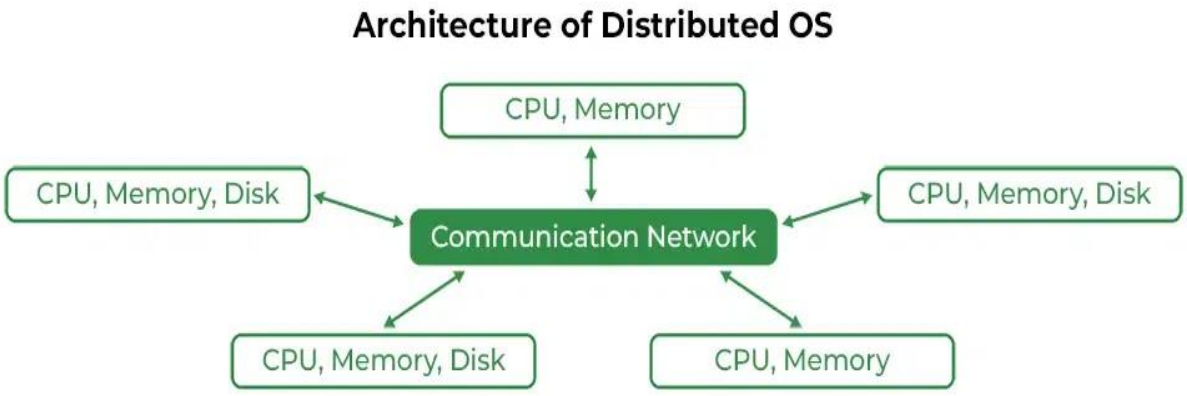
- Reliability problem.
- One must have to take care of the security and integrity of user programs and data.
- Data communication problem.
- High Overhead: Time-sharing systems have a higher overhead than other operating systems due to the need for scheduling, context switching, and other overheads that come with supporting multiple users.
- Complexity: Time-sharing systems are complex and require advanced software to manage multiple users simultaneously. This complexity increases the chance of bugs and errors.
- Security Risks: With multiple users sharing resources, the risk of security breaches increases. Time-sharing systems require careful management of user access, authentication, and authorization to ensure the security of data and software.

Examples of Time-Sharing OS with explanation

- **IBM VM/CMS**: IBM VM/CMS is a time-sharing operating system that was first introduced in 1972. It is still in use today, providing a virtual machine environment that allows multiple users to run their own instances of operating systems and applications.
- **TSO (Time Sharing Option)**: TSO is a time-sharing operating system that was first introduced in the 1960s by IBM for the IBM System/360 mainframe computer. It allowed multiple users to access the same computer simultaneously, running their own applications.
- **Windows Terminal Services**: Windows Terminal Services is a time-sharing operating system that allows multiple users to access a Windows server remotely. Users can run their own applications and access shared resources, such as printers and network storage, in real-time.

6. Distributed Operating System

These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, at a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as loosely coupled systems or distributed systems. These systems' processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



Advantages of Distributed Operating System

- Failure of one will not affect the other network communication, as all systems are independent of each other.
- Electronic mail increases the data exchange speed.
- Since resources are being shared, computation is highly fast and durable.
- Load on host computer reduces.

- These systems are easily scalable as many systems can be easily added to the network.
- Delay in data processing reduces.

Disadvantages of Distributed Operating System

- Failure of the main network will stop the entire communication.
- To establish distributed systems the language is used not well-defined yet.
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet.

Examples of Distributed Operating Systems are LOCUS, etc.

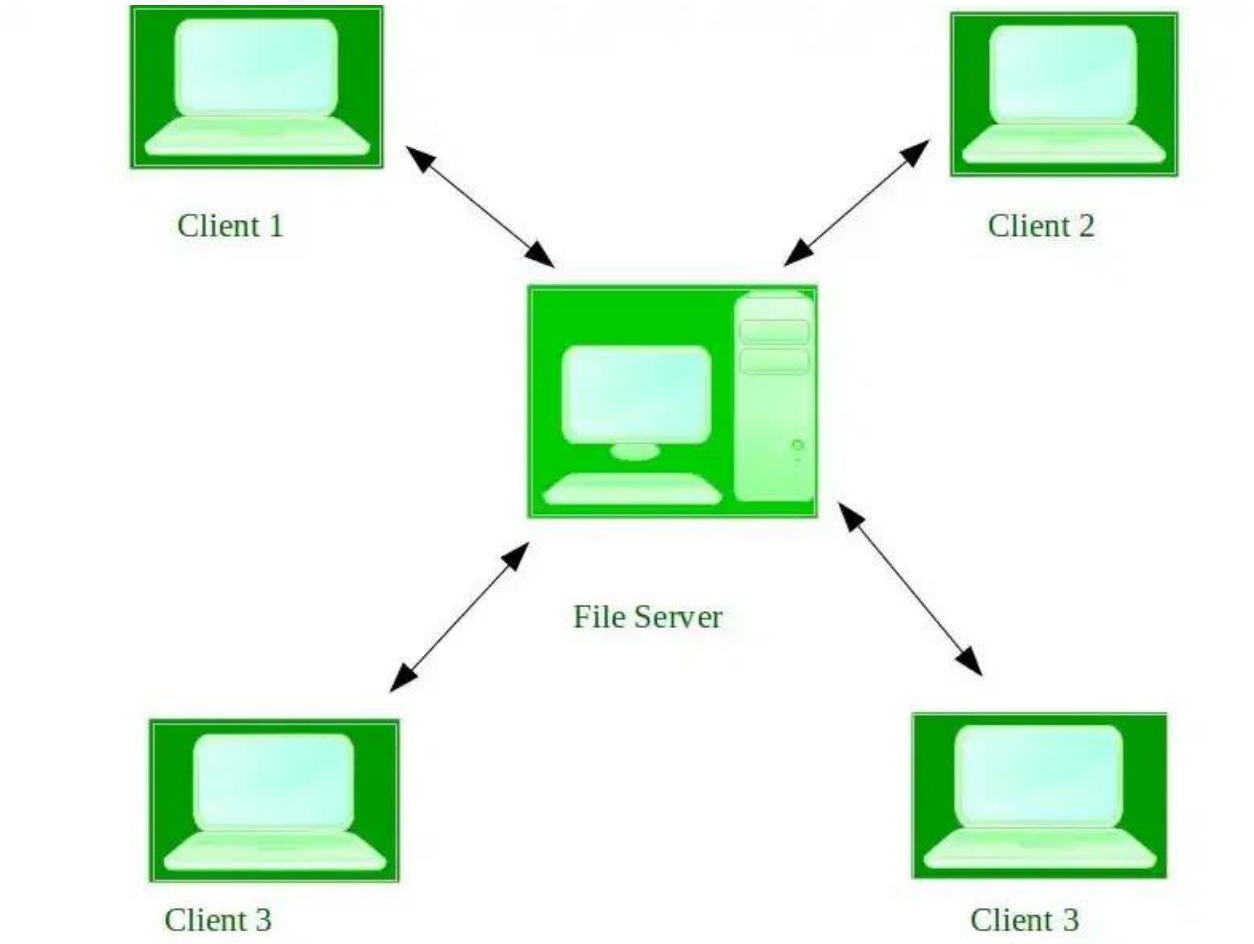
Issues With Distributed Operating Systems

- Networking causes delays in the transfer of data between nodes of a distributed system. Such delays may lead to an inconsistent view of data located in different nodes, and make it difficult to know the chronological order in which events occurred in the system.
- Control functions like scheduling, resource allocation, and deadlock detection have to be performed in several nodes to achieve computation speedup and provide reliable operation when computers or networking components fail.
- Messages exchanged by processes present in different nodes may travel over public networks and pass through computer systems that are not controlled by the distributed operating system. An intruder may exploit this feature to tamper with messages, or create fake messages to fool the authentication procedure and masquerade as a user of the system.

7. Network Operating System

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access to files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual

connections, etc. and that's why these computers are popularly known as tightly coupled systems



Advantages of Network Operating System

- Highly stable centralized servers.
- Security concerns are handled through servers.
- New technologies and hardware up-gradation are easily integrated into the system.
- Server access is possible remotely from different locations and types of systems.

Disadvantages of Network Operating System

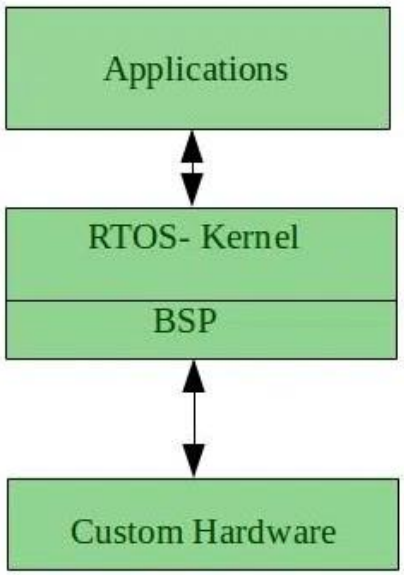
- Servers are costly.

- User has to depend on a central location for most operations.
- Maintenance and updates are required regularly.

Examples of Network Operating Systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, BSD, etc.

8. Real-Time Operating System

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**. **Real-time systems** are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.



Types of Real-Time Operating Systems

- **Hard Real-Time Systems:** Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily

available in case of an accident. Virtual memory is rarely found in these systems.

- **Soft Real-Time Systems:** These OSs are for applications where time-constraint is less strict.

For more, refer to the Difference Between Hard Real-Time OS and Soft Real-Time OS.

Examples of Real-Time Operating Systems are Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Advantages of RTOS

- **Maximum Consumption:** Maximum utilization of devices and systems, thus more output from all the resources.
- **Task Shifting:** The time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds in shifting from one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance on applications that are in the queue.
- Real-time **operating system in** the **embedded system:** Since the size of programs is small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages of RTOS

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on a few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signal to respond earliest to interrupts.

- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

9. Multithreading Operating System

Multithreading is a feature in operating systems that allows a program to do several tasks at the same time. Think of it like having multiple hands working together to complete different parts of a job faster. Each "hand" is called a thread, and they help make programs run more efficiently. Multithreading makes your computer work better by using its resources more effectively, leading to quicker and smoother performance for applications like web browsers, games, and many other programs you use every day.

How Does Multithreading Work?

Multithreading works by allowing a computer's processor to handle multiple tasks at the same time. Even though the processor can only do one thing at a time, it switches between different threads from various programs so quickly that it looks like everything is happening all at once.

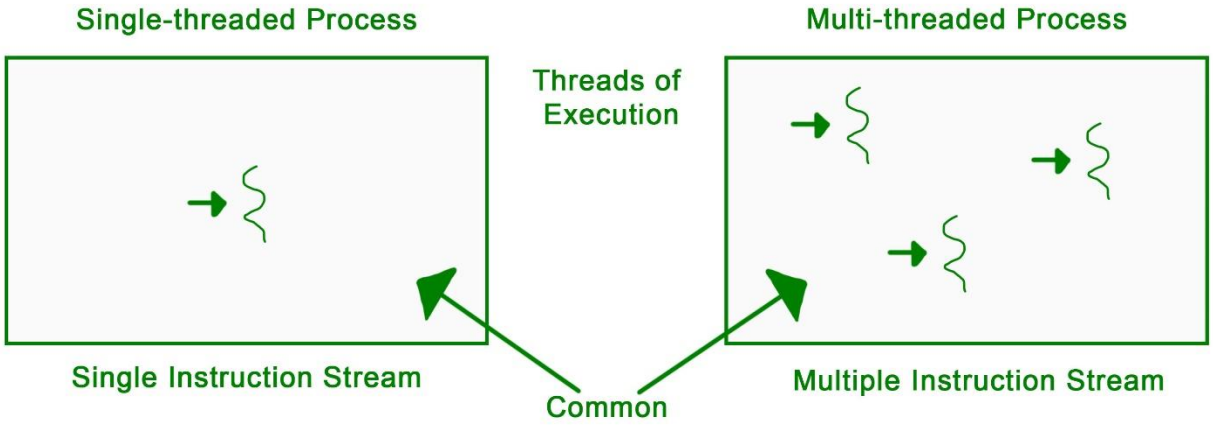
Here's how it simplifies:

- **Processor Handling**: The processor can execute only one instruction at a time, but it switches between different threads so fast that it gives the illusion of simultaneous execution.
- **Thread Synchronization**: Each thread is like a separate task within a program. They share resources and work together smoothly, ensuring programs run efficiently.
- **Efficient Execution**: Threads in a program can run independently or wait for their turn to process, making programs faster and more responsive.
- **Programming Considerations**: Programmers need to be careful about managing threads to avoid problems like conflicts or situations where threads get stuck waiting for each other.

What is Multitasking?

Multitasking is the ability of an operating system to run multiple programs or tasks at the same time. It allows you to perform different activities simultaneously on your computer. For example, you can listen to music while browsing the internet and typing a document all at once.

Multitasking is of two types: Processor-based and thread-based. Processor-based multitasking is managed by the OS, however, multitasking through multithreading can be controlled by the programmer to some extent. The concept of **multi-threading** needs a proper understanding of these two terms – **a process and a thread**. A process is a program being executed. A process can be further divided into independent units known as threads. A thread is like a small light-weight process within a process. Or we can say a collection of threads is what is known as a process.



Single Thread and Multi Thread Process

Applications: Threading is used widely in almost every field. Most widely it is seen over the internet nowadays where we are using transaction processing of every type like recharges, online transfer, banking etc. Threading is a segment which divide the code into small parts that are of very light weight and has less burden on CPU memory so that it can be easily worked out and can achieve goal in desired field. The concept of threading is designed due to the problem of fast and regular changes in technology and less the work in different areas due to less application. Then as says "need is the generation of creation or innovation" hence by following this approach human mind develop the concept of thread to enhance the capability of programming.

Multithreading vs Multitasking

Feature	Multithreading	Multitasking
Definition	Running multiple threads within a single program simultaneously.	Running multiple programs or tasks concurrently.
Example	Web browser loading a page, handling user input, and downloading files simultaneously.	Listening to music, browsing the web, and typing a document at the same time.
Scope	Within a single program.	Across multiple programs.
Resource Use	Utilizes CPU resources more efficiently within a program.	Manages system resources to allocate time and memory to different programs.
Purpose	Enhances the performance and responsiveness of a single application.	Improves overall system efficiency by allowing concurrent execution of multiple programs.
Switching	Threads are managed by the program itself.	Programs are managed by the operating system, which switches between them.

Lifecycle of a Thread

There are various stages in the lifecycle of a thread. Following are the stages a thread goes through in its whole life.

- **New:** The lifecycle of a born thread (new thread) starts in this state. It remains in this state till a program starts.
- **Runnable**: A thread becomes runnable after it starts. It is considered to be executing the task given to it.
- **Waiting**: While waiting for another thread to perform a task, the currently running thread goes into the waiting state and then transitions back again after receiving a signal from the other thread.
- **Timed Waiting:** A runnable thread enters into this state for a specific time interval and then transitions back when the time interval expires or the event the thread was waiting for occurs.
- **Terminated (Dead)**: A thread enters into this state after completing its task.

Types of Execution in OS

There are two types of execution:

- **Concurrent Execution:** This occurs when a processor is successful in switching resources between threads in a multithreaded process on a single processor.
- **Parallel Execution:** This occurs when every thread in the process runs on a separate processor at the same time and in the same multithreaded process

Drawbacks of Multithreading

Multithreading is complex and many times difficult to handle. It has a few drawbacks. These are:

- If you don't make use of the locking mechanisms properly, while investigating data access issues there is a chance of problems arising like data inconsistency and dead-lock.
- If many threads try to access the same data, then there is a chance that the situation of thread starvation may arise. Resource contention issues are another problem that can trouble the user.
- Display issues may occur if threads lack coordination when displaying data.

Benefits of Multithreading

- Multithreading can improve the performance and efficiency of a program by utilizing the available CPU resources more effectively. Executing multiple threads concurrently, it can take advantage of parallelism and reduce overall execution time.
- Multithreading can enhance responsiveness in applications that involve user interaction. By separating time-consuming tasks from the main thread, the user interface can remain responsive and not freeze or become unresponsive.
- Multithreading can enable better resource utilization. For example, in a server application, multiple threads can handle incoming client requests simultaneously, allowing the server to serve more clients concurrently.
- Multithreading can facilitate better code organization and modularity by dividing complex tasks into smaller, manageable units of execution. Each thread can handle a specific part of the task, making the code easier to understand and maintain.