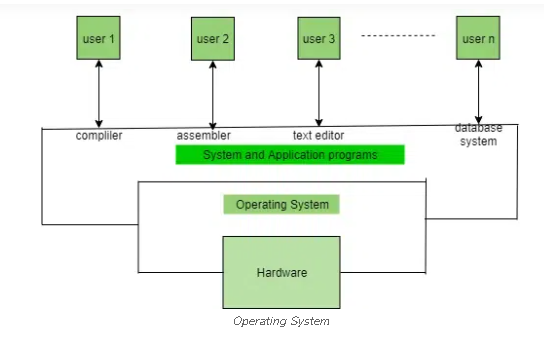
Chapter-1

|  |
| --- |
| **Introduction:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems. |
| OS Services, System calls Structure of an OS-Layered, |
| Monolithic, Microkernel Operating Systems, Concept of Virtual Machine |

**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.



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)](https://www.geeksforgeeks.org/difference-between-cli-and-gui/), 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 error messages, and other debugging and error-detecting methods.

**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.

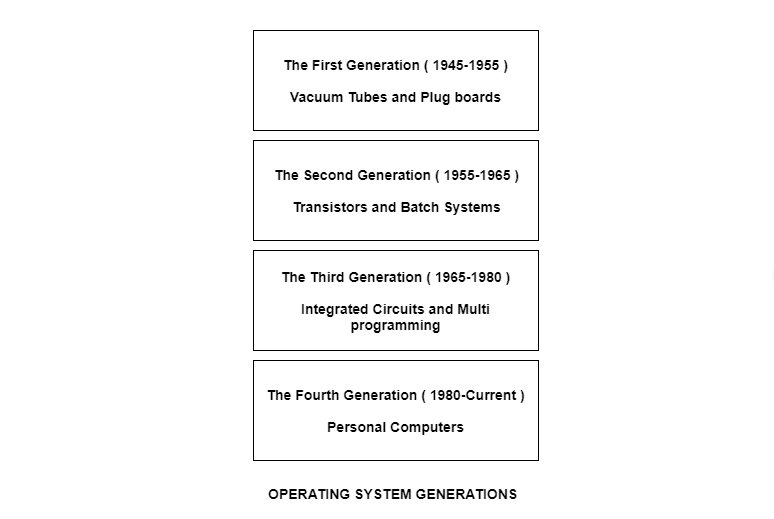
**Easy Access:** To provide easy access to users for using resources by acting as an intermediary between the hardware and its users**.**

**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.

[**Operating Systems**](https://www.tutorialspoint.com/operating_system/index.htm) have evolved over the years. So, their evolution through the years can be mapped using generations of operating systems. There are four generations of operating systems. These can be described as follows −



## The First Generation ( 1945 - 1955 ): Vacuum Tubes and Plugboards

Digital computers were not constructed until the second world war. Calculating engines with mechanical relays were built at that time. However, the mechanical relays were very slow and were later replaced with vacuum tubes. These machines were enormous but were still very slow.

These early computers were designed, built and maintained by a single group of people. Programming languages were unknown and there were no operating systems so all the programming was done in machine language. All the problems were simple numerical calculations.

By the 1950’s punch cards were introduced and this improved the computer system. Instead of using plugboards, programs were written on cards and read into the system.

## The Second Generation ( 1955 - 1965 ): Transistors and Batch Systems

[**Transistors**](https://www.tutorialspoint.com/basic_electronics/basic_electronics_transistors.htm) led to the development of the computer systems that could be manufactured and sold to paying customers. These machines were known as mainframes and were locked in air-conditioned computer rooms with staff to operate them.

The [**Batch System**](https://www.tutorialspoint.com/batch-operating-system) was introduced to reduce the wasted time in the computer. A tray full of jobs was collected in the input room and read into the [**magnetic tape**](https://www.tutorialspoint.com/what-is-magnetic-tape). After that, the tape was rewound and mounted on a tape drive. Then the batch operating system was loaded in which read the first job from the tape and ran it. The output was written on the second tape. After the whole batch was done, the input and output tapes were removed and the output tape was printed.

## The Third Generation ( 1965 - 1980 ): Integrated Circuits and Multiprogramming

Until the 1960’s, there were two types of computer systems i.e., the scientific and the commercial computers. These were combined by IBM in the System/360. This used integrated circuits and provided a major price and performance advantage over the second generation systems.

The third generation operating systems also introduced [**multiprogramming**](https://www.tutorialspoint.com/what-is-a-multiprogramming-operating-system). This meant that the processor was not idle while a job was completing its I/O operation. Another job was scheduled on the processor so that its time would not be wasted.

## The Fourth Generation ( 1980 - Present ): Personal Computers

Personal Computers were easy to create with the development of large-scale integrated circuits. These were chips containing thousands of transistors on a square centimeter of silicon. Because of these, microcomputers were much cheaper than minicomputers and that made it possible for a single individual to own one of them.

The advent of personal computers also led to the growth of networks. This created network operating systems and distributed operating systems. The users were aware of a network while using a network operating system and could log in to remote machines and copy files from one machine to another.

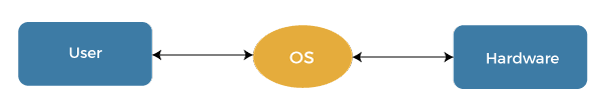
# **Operating system services**

In this article, we understand the basics of an operating system and its services.

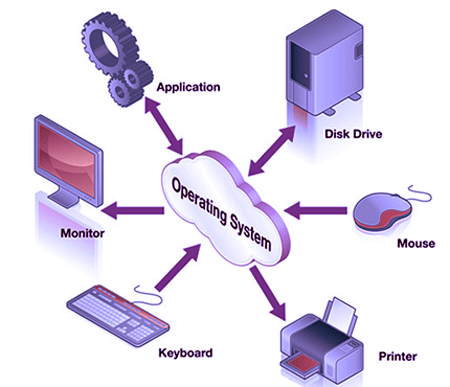
### **What do you mean by Operating system?**

No matter its size and application, every computer needs an operating system to make it functional and useful. The operating system is an integral part of modern computer systems. It is a well-organized collection of programs that manages the hardware.

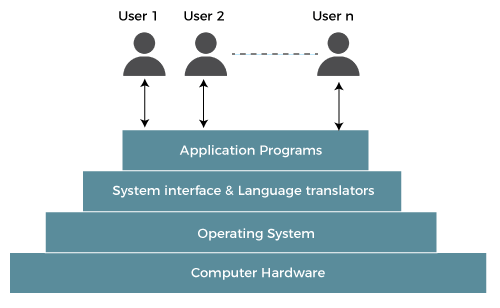
An Operating System provides an interaction between the users and computer hardware. A user is a person sitting at the computer terminal concerned about the application rather than the architecture of the computer. The user never interacts with the hardware directly. To get the services of the hardware, he has to request through the operating system.



The operating system is a primary resource manager. It manages the hardware, including processors, memory, Input-Output devices, and communication devices.



The operating system operates either in kernel mode or user mode. Compilers and editors run in user mode, whereas operating system code runs in kernel mode.



**Operating system services:**

The operating system provides the programming environment in which a programmer works on a computer system. The user program requests various resources through the operating system. The operating system gives several services to utility programmers and users. Applications access these services through application programming interfaces or system calls. By invoking those interfaces, the application can request a service from the operating system, pass parameters, and acquire the operation outcomes.

**Following are the services provided by an operating system -**

* Program execution
* Control Input/output devices
* Program creation
* Error Detection and Response
* Accounting
* Security and Protection
* File Management
* Communication

**Program execution**

To execute a program, several tasks need to be performed. Both the instructions and data must be loaded into the main memory. In addition, input-output devices and files should be initialized, and other resources must be prepared. The Operating structures handle these kinds of tasks. The user now no longer should fear the reminiscence allocation or multitasking or anything.

**Control Input/output devices**

As there are numerous types of I/O devices within the computer system, and each I/O device calls for its own precise set of instructions for the operation. The Operating System hides that info with the aid of presenting a uniform interface. Thus, it is convenient for programmers to access such devices easily.

**Program Creation**

The Operating system offers the structures and tools, including editors and debuggers, to help the programmer create, modify, and debugging programs.

**Error Detection and Response**

An Error in a device may also cause malfunctioning of the entire device. These include hardware and software errors such as device failure, memory error, division by zero, attempts to access forbidden memory locations, etc. To avoid error, the operating system monitors the system for detecting errors and takes suitable action with at least impact on running applications.

While working with computers, errors may occur quite often. Errors may occur in the:

* **Input/ Output devices:** For example, connection failure in the network, lack of paper in the printer, etc.
* **User program:** For example: attempt to access illegal memory locations, divide by zero, use too much CPU time, etc.
* **Memory hardware:** For example, Memory error, the memory becomes full, etc.

To handle these errors and other types of possible errors, the operating system takes appropriate action and generates messages to ensure correct and consistent computing.

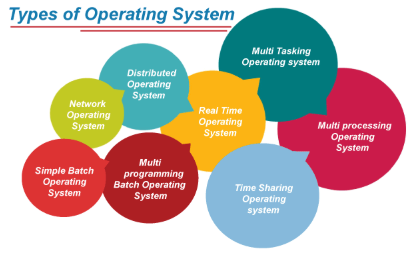
**Accounting**

An Operating device collects utilization records for numerous assets and tracks the overall performance parameters and responsive time to enhance overall performance. These personal records are beneficial for additional upgrades and tuning the device to enhance overall performance.

**Security and Protection**

Operating device affords safety to the statistics and packages of a person and protects any interference from unauthorized users. The safety feature counters threats, which are published via way of individuals out of doors the manage of the running device.

**Types of Operating Systems**



**Batch Operating System:**A [Batch Operating System](https://www.geeksforgeeks.org/batch-processing-operating-system/) is a type of operating system that does not interact with the computer directly. There is an operator who takes similar jobs having the same requirements and groups them into batches.

**Time-sharing Operating System:**[Time-sharing Operating System](https://www.geeksforgeeks.org/time-sharing-operating-system/) is a type of operating system that allows many users to share computer resources (maximum utilization of the resources).

**Distributed Operating System:**[Distributed Operating System](https://www.geeksforgeeks.org/what-is-a-distributed-system/) is a type of operating system that manages a group of different computers and makes appear to be a single computer. These operating systems are designed to operate on a network of computers. They allow multiple users to access shared resources and communicate with each other over the network. Examples include Microsoft Windows Server and various distributions of Linux designed for servers.

**Network Operating System:**[Network Operating System](https://www.geeksforgeeks.org/what-is-a-network-operating-system/) is a type of operating system that runs on a server and provides the capability to manage data, users, groups, security, applications, and other networking functions.

**Real-time Operating System:**[Real-time Operating System](https://www.geeksforgeeks.org/real-time-operating-system-rtos/) is a type of operating system that serves a real-time system and the time interval required to process and respond to inputs is very small. These operating systems are designed to respond to events in real time. They are used in applications that require quick and deterministic responses, such as embedded systems, industrial control systems, and robotics.

**Multiprocessing Operating System:**[Multiprocessor Operating Systems](https://www.geeksforgeeks.org/introduction-of-multiprocessor-and-multicomputer/) are used in operating systems to boost the performance of multiple CPUs within a single computer system. Multiple CPUs are linked together so that a job can be divided and executed more quickly.

**Single-User Operating Systems:**[Single-User Operating Systems](https://www.geeksforgeeks.org/single-user-operating-system/) are designed to support a single user at a time. Examples include Microsoft Windows for personal computers and Apple macOS.

**Multi-User Operating Systems:**[Multi-User Operating Systems](https://www.geeksforgeeks.org/multi-user-operating-system/) are designed to support multiple users simultaneously. Examples include Linux and Unix.

**Embedded Operating Systems:**[Embedded Operating Systems](https://www.geeksforgeeks.org/embedded-real-time-system/)are designed to run on devices with limited resources, such as smartphones, wearable devices, and household appliances. Examples include Google’s Android and Apple’s iOS.

**Cluster Operating Systems:** Cluster Operating Systems are designed to run on a group of computers, or a cluster, to work together as a single system. They are used for high-performance computing and for applications that require high availability and reliability. Examples include Rocks Cluster Distribution and OpenMPI.

Examples of Operating Systems

**Windows**(GUI-based, PC)

**GNU/Linux** (Personal, Workstations, ISP, File, and print server, Three-tier client/Server)

**macOS** (Macintosh), used for Apple’s personal computers and workstations (MacBook, iMac).

**Android** (Google’s Operating System for smartphones/tablets/smartwatches)

**iOS** (Apple’s OS for iPhone, iPad, and iPod Touch)

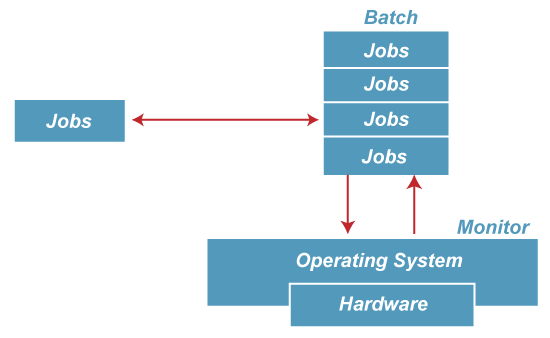
## Types of Operating Systems

### **1. Batch Operating System**

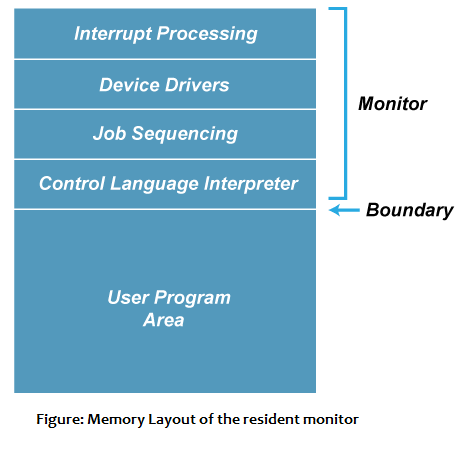
In the 1970s, Batch processing was viral. In this technique, similar types of jobs were batched together and executed in time. People were used to having a single computer, which was called a mainframe.

In the Batch operating system, access is given to more than one person; they submit their respective jobs to the system for execution.

The system puts all of the jobs in a queue on the basis of first come, first serve and then executes the jobs one by one. The users collect their respective output when all the jobs get executed.



The purpose of this operating system was mainly to transfer control from one job to another as soon as the job was completed. It contained a small set of programs called the resident monitor that always resided in one part of the main memory. The remaining part is used for servicing jobs.



### **Advantages of Batch OS**

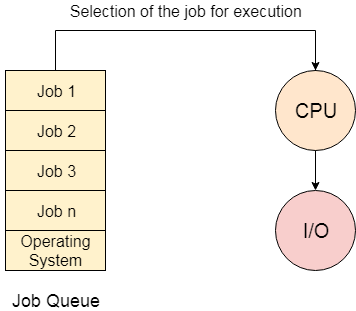
* It has increased efficiency, like a resident monitor that eliminates CPU idle time between jobs to enable it to smoothen the switching of tasks.
* The automatic handling of the processing of jobs means that users should not take control to intervene in whatever is going on while running or executing tasks.
* The resources, such as the CPU and the memory, are utilized and not left idle.
* It goes well with the payroll as well because it keeps processing jobs in batches without any time wastage and hard work.
* It can also perform even very huge complex jobs without any intermission because it simply streams without any intermission, even when the task is excessively intensive.
* Group jobs as 'batches' remove most of the manual setup that had to be done in between individual tasks, thereby saving time.
* It logs and deals with errors at the time when the batch is over. This allows the system to run even without intermissions.

### **Disadvantages of Batch OS**

**1. Starvation**

Batch processing suffers from starvation.

**For Example:**



There are five jobs J1, J2, J3, J4, and J5, present in the batch. If the execution time of J1 is very high, then the other four jobs will never be executed, or they will have to wait for a very long time. Hence the other processes get starved.

**2. Not Interactive**

Batch Processing is not suitable for jobs that are dependent on the user's input. If a job requires the input of two numbers from the console, then it will never get it in the batch processing scenario since the user is not present at the time of execution.

**3. Delayed Output**

Since the jobs are submitted in batches, the output is not produced in time. Such a condition can be rather inconvenient for time-critical jobs.

**4. Difficult to Debug**

An error is found only after the entire batch has been processed, which makes it even harder to locate and fix an issue in real-time.

**5. It Requires Knowledge of Job Scheduling**

The users or the system administrator should know well about the behavior of the system as well as dependencies among tasks.

Advertisement

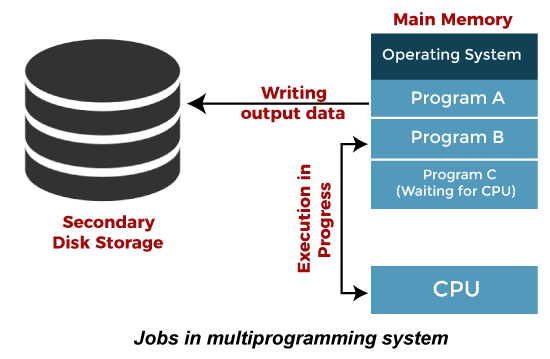
**6. Large Jobs Cause Delays**

If a batch contains a large job, then problems may occur because the processing of all the subsequent jobs is delayed. This, therefore, slows down the overall system performance.

## 2. Multiprogramming Operating System

Multiprogramming is an extension to batch processing where the CPU is always kept busy. Each process needs two types of system time: CPU time and IO time.

In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes. Therefore, multiprogramming improves the efficiency of the system.



### **Advantages of Multiprogramming OS**

* Throughout the system, it increased as the CPU always had one program to execute.
* Response time can also be reduced.
* Multiprogramming maximizes the utilization of resources like memory, I/O devices, and processing power since more than one program can be kept alive at any time.
* Since several jobs are being processed in parallel, significantly more tasks could be completed within a certain amount of time, thus enhancing the overall throughput of the system.
* During times when a program is waiting for I/O operations, the processor does not go idle since it jumps on to another task to continue processing.
* The system can support both short and long tasks to be executed in parallel, which makes for a more dynamic and productive processing environment.

### **Disadvantages of Multiprogramming OS**

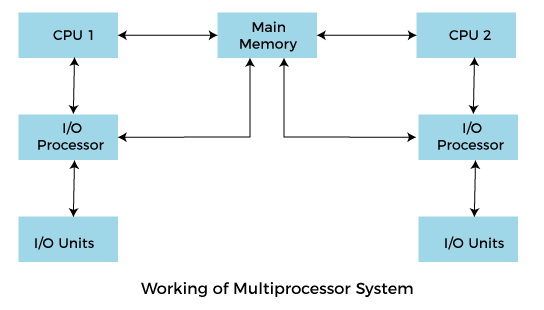
* Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.
* Multiple programs increase system complexity, as the operating system needs to manage multiple processes, memory management, and scheduling.
* It requires more memory compared to less sophisticated operating systems because multiple programs run in memory simultaneously.
* The operating system has to switch between the running processes continuously, and this leads to scheduling overhead and reduces performance.
* As several concurrent operations access shared resources simultaneously, the system is likely to experience deadlocks: two or more processes waiting for each other for further actions.
* Resource contention due to competition for scarce resources could degrade performance.

## 3. Multiprocessing Operating System

In Multiprocessing, Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time. This will increase the throughput of the system.

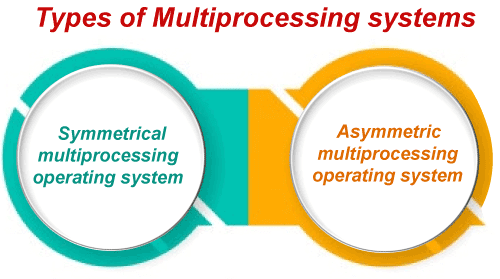
* In multiprocessing, several processors execute procedures in collaboration. You can even split a procedure into parts that run in parallel.
* Since several processors work concurrently, more tasks are completed faster.
* Multiprocessing enhances the speed at which the system executes since tasks scatter the processors, minimizing the waiting period for procedures.
* If one processor is faulty, the others will continue running, hence enhancing system reliability.

Such an operating system is suited for highly computationally intensive environments needing high processing power, for example, scientific simulations, real-time data processing, and server environments



**Working**

In a Multiprocessing OS, concurrent execution of processes by multiple processors is done. The processes are divided and allotted to different processors for parallel computing. Thus, system performance is enhanced due to high throughput, good resource allocation, smooth distribution of workload on the processors, and fault tolerance. If one of its processors fails, then the system can continue and reallocate its tasks to other processors, and thus, the operation can be done smoothly.



There are two types of Multiprocessing Operating Systems:

**i. Symmetric Multiprocessing:**

* In SMP, every processor has the same copy of the operating system; they share memory and I/O devices. Each processor can do any task; it would ensure that the workload is balanced among all of them.
* Each processor can continue any work; if one fails or stops functioning, others can continue without affecting the system's performance.

**ii. Asymmetric Multiprocessing:**

* In AMP, one processor controls the system. It manages what tasks are to be given to the other processors.
* The master processor provides service to the operating system, whereas other processors are slaves and may perform specific work for the system.
* This configuration of the system keeps it very simple but sometimes causes bottlenecks if the master processor gets too loaded.

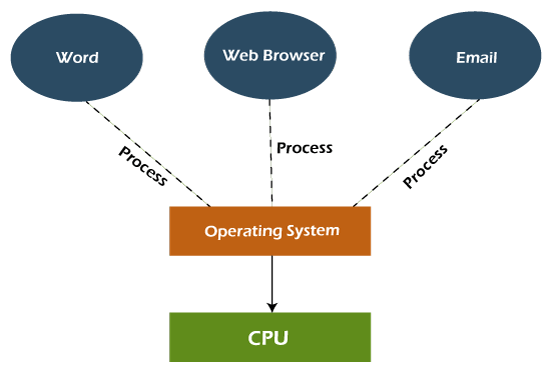
### **Advantages of Multiprocessing Operating System:**

* Increased Reliability: Because multiple processors are present, if one fails, others may take over it, thus stabilizing the system.
* Increased Throughput: More jobs can be processed at the same time by these multiple processors than by a single processor, thereby increasing the speed of execution.
* Efficient Resource Utilization: It acts more productively upon the utilization of resources like CPU, memory, and I/O devices.
* Parallelism: Many processes can run parallel, which increases the speed of execution to a large extent.
* Scalability: As the workload increases, more processors can be added to enhance performance.

### **Disadvantages of Multiprocessing Operating System:**

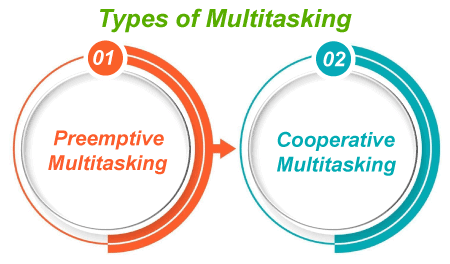
* **Complexity:** There is a rise in the complexity of the system while dealing with a number of processors and proper distribution of tasks.
* **Increased Cost:** Hardware equipment added in multiprocessing systems increases their cost
* **ommunication Overhead:** Generally, communication among processors results in overhead and provides a slight reduction in efficiency.
* **Software Compatibility Problems:** Generally, most software is not designed to operate properly with multiprocessing systems.

### **4. Multitasking Operating System**



Multiprogramming extended the multitasking operating system concept, which enabled various programs to run simultaneously. It provided a user with the ability to do more than one task on the same system at a given time by sharing various system resources like CPU time among different processes. This assures efficient use of the CPU because it shifts between tasks quickly; hence, it seems that they are running parallel.

There are multitasking systems in almost all personal computers where, for example, a user might run a web browser, a music player, and a document editor all at once.



**i. Cooperative Operating System:**

In a co-operating system, the tasks voluntarily relinquish control of the CPU so other tasks can execute. A process runs until it completes or explicitly relinquishes control. An operating system has a relatively small amount of control over when a process will context switch. The operating system must rely on the processes to co-operate with each other. Such a system may lead to problems if a process does not yield control and causes others to be blocked.

**ii. Preemptive Operating System:**

n this system, the operating system has full control over how it allocates the CPU. In this system, it can interrupt the process in the middle of its execution and switch to another based on previously defined criteria for scheduling such as priority, time slices, or deadlines. This means that within such a system, better multitasking and responsiveness are ensured, and there is no event of allowing any single process to monopolize the CPU.

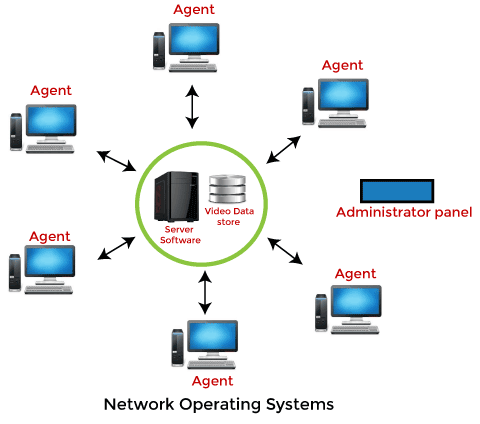
### **Advantages of the Multitasking Operating System:**

* This system can handle multiple users or tasks at once; therefore, it is best utilized in multi-user environments.
* The memory is allocated dynamically and efficiently to various tasks so that there is optimal usage of system resources.
* More applications can be run at the same time, which increases productivity because tasks are executed concurrently.
* It provides for rapid shifts between tasks, therefore, shortening the response time for the user.
* Because the system keeps running by switching between tasks, CPU time and other system resources are consumed more effectively.

### **Disadvantages of Multitasking Operating System:**

* The multitasking environment makes a number of processors busier at the same time; that is, the CPU generates more heat.
* The managing of different tasks all together requires a more sophisticated algorithm, which becomes complex to administer.
* Running too many applications at the same time can stress the system to a point where performance degrades.
* In this scenario, multiple tasks will be competing for the same resources, thereby delaying them.
* Multitasking systems usually require more powerful hardware, especially in terms of memory and processing power, to run without a hitch.

## 5. Network Operating System

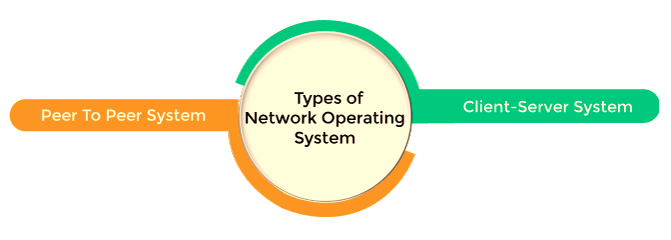


A NOS, which is an abbreviation for network operating system, is an operating system that exists to allow computers to communicate and share resources over a network, almost like the backbone of networking, ensuring that different devices (like computers, printers, and even servers) can always connect and interact seamlessly.

It's very illustrative to imagine a situation in which the students share the same whiteboard-a shared resource like a printer or file and can't all go at once. The network operating system is then like the teacher, keeping everything organized so that every student gets their turn without chaos. Other things could be controlling data sharing, security, and communication among devices.

NOS allows devices not only to communicate but also to share software applications, files, and hardware so everyone in the network can have what they need without a complicated setup. Examples of network operating systems are Microsoft Windows Server, Linux, and Novell NetWare.

The enterprise saves money since resources like printers or storage drives do not have to be duplicated for every user as they are shared, making the system in its whole aspect easier and cheaper to operate. It is the tool that keeps the network running smoothly, making it possible for connected devices to work together in collaboration and productivity.



### **Advantages of Network Operating Systems**

* Since network applications are split between the clients and the servers, this minimizes the total number of communications on the network, thus increasing performance.
* Configuring as well as maintaining an NOS is less expensive than other elaborate systems since shared resources reduce duplication needed.
* A NOS grants centralized control over data, security, and resource management. That means that administrations may easily manage a large network.
* Scaling of the systems can be made easily in terms of organization growth. Easly, new clients or servers can be added without reconfiguring the system.
* It shares resources like printers, files, and applications, which reduces hardware and software redundancy.

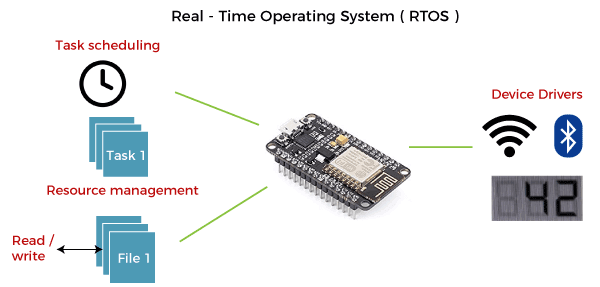
### **Disadvantage of Network Operating System**

* If one node or server fails, it affects the whole system, and network functions will be interrupted. Hence, reliability is important.
* Security needs to be robust so that unauthorized access is restricted. Complex security measures demand constant monitoring and updating.
* The skilled network administrators involved will handle system performance, security configurations, and troubleshooting.
* When the network size is too large, and traffic is heavy, it degrades with time if it's not monitored and maintained, which is constantly demanding attention.
* This is because an attacker, once able to gain access to one server, comes close to achieving his goal for multiple resources contained in the entire network.

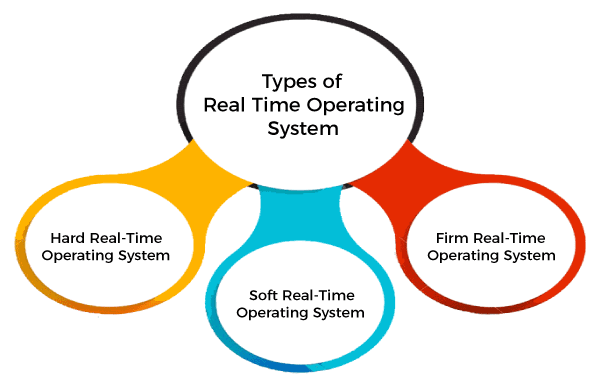
## 6. Real Time Operating System

A Real-Time Operating System is conceived as a design for tasks that need to be completed under very strict time constraints. In such systems, each task has a clear deadline; failing to achieve completion by that deadline can, in turn, bring about significant consequences. Completion by the deadline may result in either a critical failure or something no longer of relevance, again depending on the application, thus leading to inefficiencies or even dangers in its application.

Real-time systems are widely used in areas where timing is the issue, for example, the manufacturing of medical equipment, automotive controls, or even industrial automation. For instance, in an anti-lock braking system (ABS) in a car, the response must come in milliseconds in order to assure safety.



The application of a Real-Time system exists in the case of military applications. If you want to drop a missile, then the missile is supposed to be dropped with a certain precision.



**i. Hard Real-Time Systems**

In hard real-time systems, it is absolutely impossible to fail to meet a deadline. The consequence of missing a deadline may be disastrous, even failing the system or loss of life.

Examples: Flight navigation and control aircraft, Medical devices, like pacemakers.

It is required that strict guarantees be established so that the jobs will definitely be executed on time. Failure can be very disastrous.

**ii. Soft Real-Time Systems**

In this system type, occasional missed deadlines may not result in catastrophic failure but can degrade the performance or usability of the system.

Examples: Video streaming (occasional delays or buffering are acceptable) and online transaction systems (like bank ATMs, where slight delays are tolerable).

The system may still be functionally operational and workable even if some deadline misses do occur, though the performance would be adversely affected. This has focused on efficiency and the removal of delays rather than absolute precision.

**iii. Firm Real-Time Systems**

In between hard and soft real-time systems, there are firm real-time systems. In that case, there is no crash, but the result of the task is worthless when it doesn't meet the deadline. There's a minimum kind of penalty, but the value of the completion of the task goes significantly down if it is not made in time.

Examples: Automated stock trading systems (late transactions lose relevance) and some e-commerce systems (for example, offers run out).

### **Advantages of Real-Time Operating System:**

* Real-time applications are quite easy to design, develop, and implement under a real-time operating system.
* Maximum utilization of devices and systems in a Real-Time Operating System.
* Fast response to events.
* High reliability in the performance of time-critical operations.
* Strict scheduling ensures the predictable execution of tasks.

### **Disadvantages of Real-Time Operating System:**

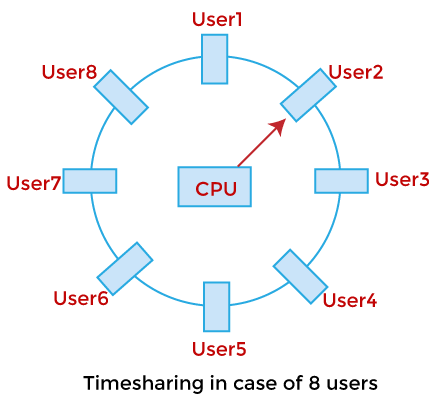
* Real-time operating systems are very expensive to design.
* Real-time operating systems are very resource-intensive and consume critical CPU cycles.
* Less multitasking support.
* Lacks adaptability to new functions.
* At times, it demands specific hardware.

## 7. Time-Sharing Operating System

A Time-Sharing Operating System allows multiple users to access the system concurrently, and this occurs by allocating a small time slice or quantum to each task. The CPU switches between the tasks so rapidly that the users feel their programs have been running concurrently. This kind of operating system is most effective in systems where multiple users require access to the central system in real-time or near real-time.

### **Here's how it works:**

* **Scheduling:** Operating systems use scheduling algorithms in order to allow a particular block of CPU time, referred to as a time slice, for each user or process. After each time slice, the CPU shifts to the following process in the queue.
* **Multiprogramming:** The TSOS uses multiprogramming, where several programs are resident in the memory at any given time. This allows several programs to be executed simultaneously.
* **Context Switching:** Whenever the CPU needs to switch to another task, the operating system saves the state of the current process and loads the state of the next process. It is a process referred to as context switching. That lets a task be resumed exactly where it left off.



A time-sharing operating system allows many users to be served simultaneously, so sophisticated CPU scheduling schemes and Input/output management are required.

Time-sharing operating systems are very difficult and expensive to build.

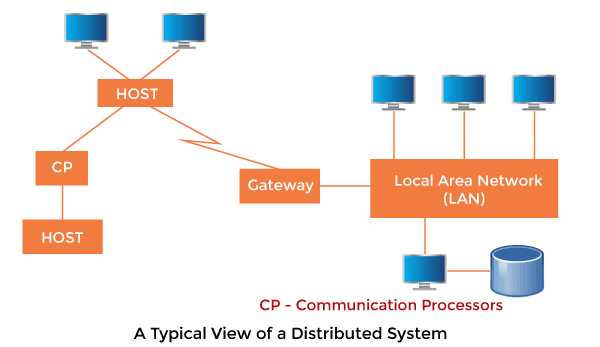
### **Advantages of Time-Sharing Operating System**

* The time-sharing operating system facilitates effective utilization and sharing of resources.
* This system helps decrease CPU idle and response time.
* It allows various users to access and interact with their programs at the same time, and it leads to greater productivity.
* Time-sharing assures better memory management because it swaps programs into and out of main memory efficiently.
* An interactive computing environment provides users with real-time access to their programs and files.

### **Disadvantages of Time-Sharing Operating System**

* Data transmission rates compared to others are very high
* The integrity and security of the user programs loaded in memory and data have to be ensured since many users access the system concurrently.
* Implementation as well as management of time-sharing systems is more complicated than others since there is a tendency for task scheduling and memory management.
* Since more and more users get hooked to the system, it degrades due to resource contention.
* However, with time, context switches between tasks will incur overheads, thereby impacting the overall efficiency of the system.

## 8. Distributed Operating System

The Distributed Operating system is not installed on a single machine; it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large, and sophisticated than Network operating systems because they also have to take care of varying networking protocols. 

### **Advantages of Distributed Operating System**

* The distributed operating system offers resource sharing.
* This is a fault-tolerant system.
* Scalability is achieved and easily new nodes can be added to the system.
* Distributed task execution improves its performance.
* Parallel processing helps in increasing the speed of job execution and enhances efficiency in getting results.

### **Disadvantages of Distributed Operating System**

* Protocol overhead might control computation costs.
* The managing of the system is complex as it works in distributed.
* Security may prove difficult on multiple nodes.
* The system highly depends on network stability for it to run without hassles.

**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 –

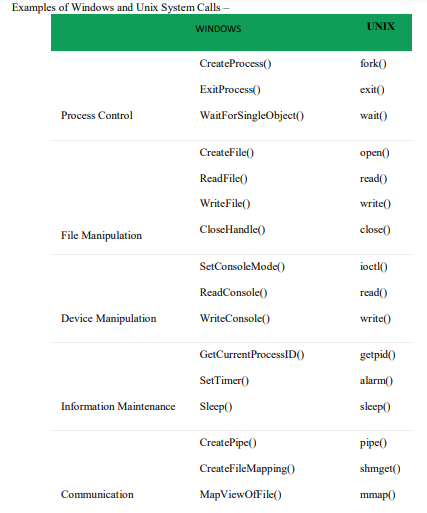
1. Process control: end, abort, create, terminate, allocate and free memory.

2. File management: create, open, close, delete, read file etc.

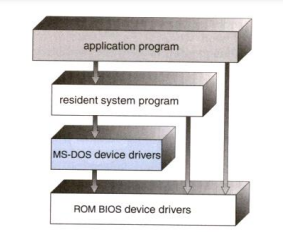
3. Device management

4. Information maintenance

5. Communication



**Operating System structure**



In MS-DOS, applications may bypass the operating system.

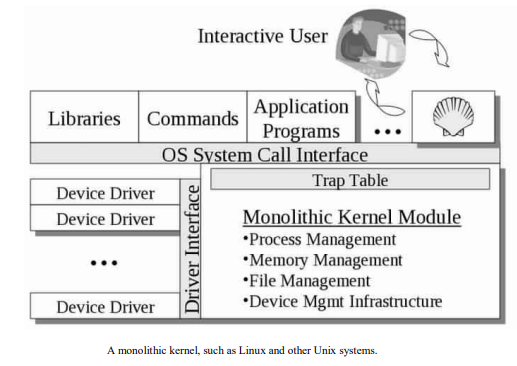
• Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.

• There was no CPU Execution Mode (user and kernel), and so errors in applications could cause the whole system to crash.

Monolithic approach

• Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.

• Device drivers are loaded into the running kernel and become part of the kernel.



3.Layered Approach This approach breaks up the operating system into different layers.

• This allows implementers to change the inner workings, and increases modularity.

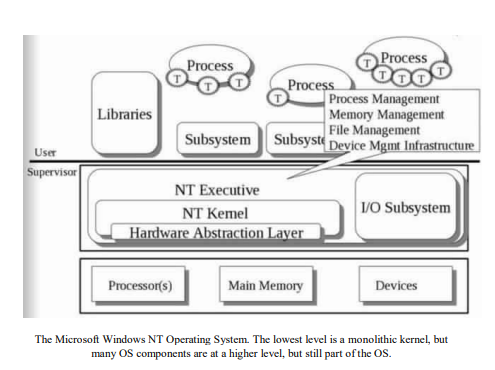
• As long as the external interface of the routines don’t change, developers have more freedom to change the inner workings of the routines.

• With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.

o The main advantage is simplicity of construction and debugging.

o The main difficulty is defining the various layers.

o The main disadvantage is that the OS tends to be less efficient than other implementations.



# What are Virtual Machines in Operating System?

A **virtual machine (VM)** is a virtual environment which functions as a virtual computer system with its own [**CPU**](https://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm), memory, network interface, and storage, created on a physical hardware system.

VMs are isolated from the rest of the system, and multiple VMs can exist on a single piece of hardware, like a server. That means, it as a simulated image of application software and operating system which is executed on a host computer or a server.

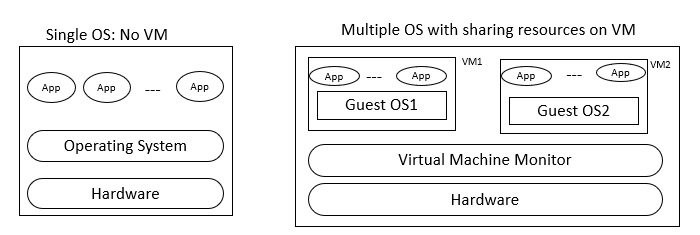
It has its own operating system and software that will facilitate the resources to virtual computers.

## Characteristics of virtual machines

The characteristics of the virtual machines are as follows −

* Multiple OS systems use the same hardware and partition resources between virtual computers.
* Separate Security and configuration identity.
* Ability to move the virtual computers between the physical host computers as holistically integrated files.

The below diagram shows you the difference between the single OS with no VM and Multiple OS with VM −



## Benefits

Let us see the major benefits of virtual machines for operating-system designers and users which are as follows −

* The multiple Operating system environments exist simultaneously on the same machine, which is isolated from each other.
* Virtual machine offers an instruction set architecture which differs from real computer.
* Using virtual machines, there is easy maintenance, application provisioning, availability and convenient recovery.

Virtual Machine encourages the users to go beyond the limitations of hardware to achieve their goals.

The operating system achieves virtualization with the help of a specialized software called a hypervisor, which emulates the PC client or server CPU, memory, hard disk, network and other hardware resources completely, enabling virtual machines to share resources.

The hypervisor can emulate multiple virtual hardware platforms that are isolated from each other allowing virtual machines to run Linux and window server operating machines on the same underlying physical host.

### **Basis for developing the OS**

Create the illusion of having one or more objects to emulate the real object. It is closely related to abstraction. In developing the OS, abstraction provides simplification by combining multiple simple objects into a single complex object

Virtualization provides diversification and replication by creating the illusion of objects with desired characteristics.