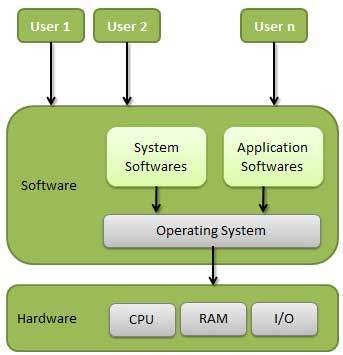
An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

An operating system is a software that manages the computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

**Operating System –** Definition:

Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Following are some of important functions of an operating System.

* Memory Management
* Processor Management
* Device Management
* File Management
* Security
* Control over system performance
* Job accounting
* Error detecting aids
* Coordination between

## Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −

* Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
* In multiprogramming, the OS decides which process will get memory when and how much.

## Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management −

* Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
* Allocates the processor (CPU) to a process.
* De-allocates processor when a process is no longer required.

## Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

* Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
* Decides which process gets the device when and for how much time.
* Allocates the device in the efficient way.
* De-allocates devices.

## File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

* Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
* Decides who gets the resources.
* Allocates the resources.

## Other Important Activities

Following are some of the important activities that an Operating System performs −

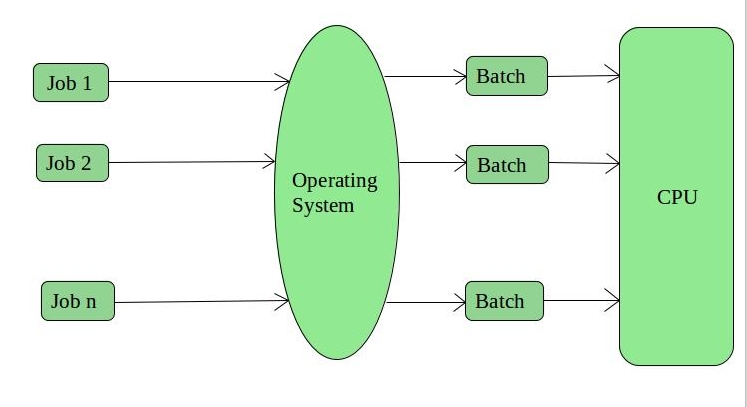
* **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
* **Control over system performance** − Recording delays between request for a service and response from the system.
* **Job accounting** − Keeping track of time and resources used by various jobs and users.
* **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
* **Coordination between other softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

# Types of Operating Systems

An [Operating System](https://www.geeksforgeeks.org/operating-system-introduction-operating-system-set-1/) performs all the basic tasks like managing file,process, and memory. Thus operating system acts as manager of all the resources, i.e. **resource manager**. Thus operating system becomes an interface between user and machine.

**Types of Operating Systems:** Some of the widely used operating systems are as follows-

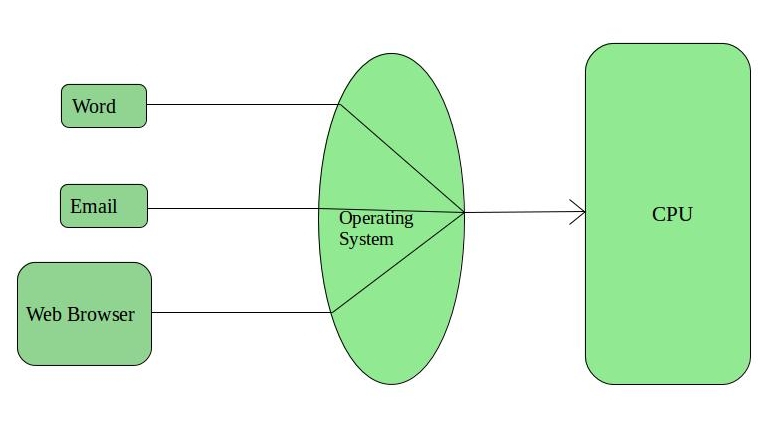
1. **Batch Operating System –**  
   This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



**Advantages of Batch Operating System:**

* It is very difficult to guess or know the time required by any job to complete. Processors of the batch systems know how long the job would be when it is in queue
* Multiple users can share the batch systems
* The idle time for batch system is very less
* It is easy to manage large work repeatedly in batch systems

**Disadvantages of Batch Operating System:**

* The computer operators should be well known with batch systems
* Batch systems are hard to debug
* It is sometime costly
* The other jobs will have to wait for an unknown time if any job fails
* **Examples of Batch based Operating System:** Payroll System, Bank Statements etc.
* **2. Time-Sharing Operating Systems –**  
  Each task is given some time to execute, so that all the tasks work smoothly. Each user gets time of CPU as they use single system. These systems are also known as Multitasking Systems. The task can be from single user or from different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to next task.

**Advantages of Time-Sharing OS:**

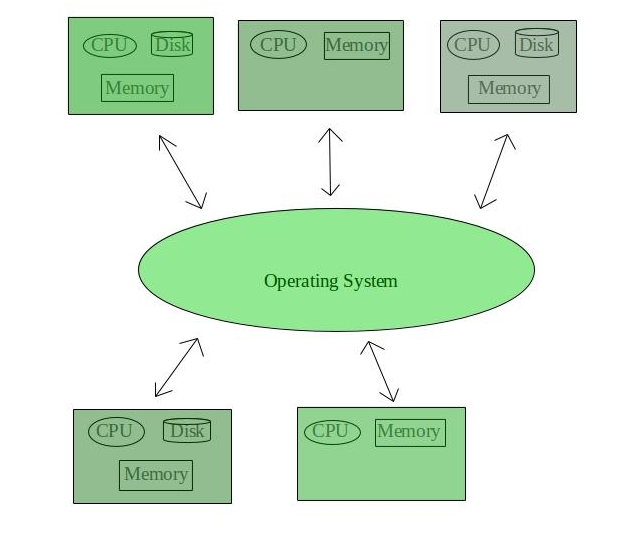
* Each task gets an equal opportunity
* Less chances of duplication of software
* CPU idle time can be reduced

**Disadvantages of Time-Sharing OS:**

* Reliability problem
* One must have to take care of security and integrity of user programs and data
* Data communication problem

**Examples of Time-Sharing OSs are:** Multics, Unix etc.

**3. Distributed Operating System –**  
These types of operating system is a recent advancement in the world of computer

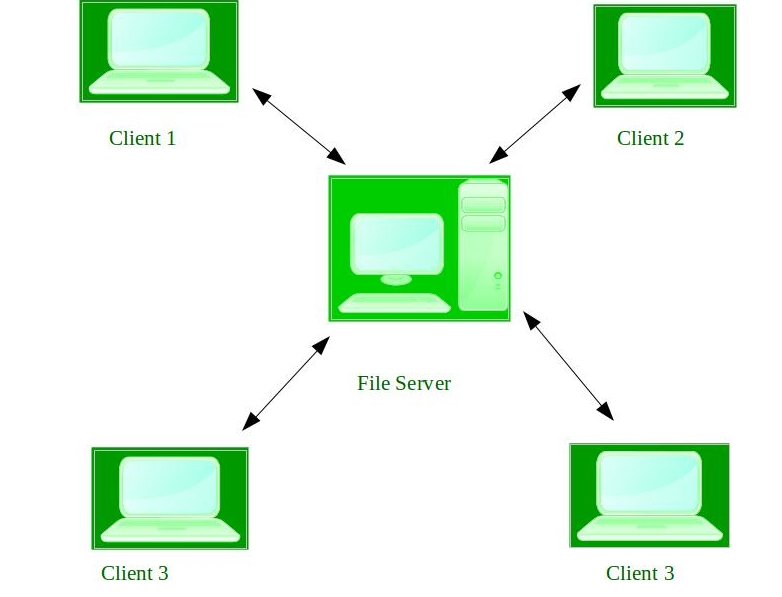
* technology and are being widely accepted all-over the world and, that too, with a great pace. Various autonomous interconnected computers communicate each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred as **loosely coupled systems** or distributed systems. These system’s processors differ in size and function. The major benefit of working with these types of 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 on 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 from 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 which are used are 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 y
* **4. 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 type of operating systems allow shared access of 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:**

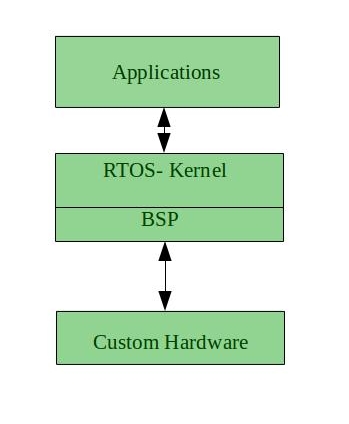
* Highly stable centralized servers
* Security concerns are handled through servers
* New technologies and hardware up-gradation are easily integrated to the system
* Server access are possible remotely from different locations and types of systems

**Disadvantages of Network Operating System:**

* Servers are costly
* User has to depend on central location for most operations
* Maintenance and updates are required regularly

**Examples of Network Operating System are:** Microsoft Windows Server 2003,

**5. Real-Time Operating System –**  
These types of OSs serves the 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 are very strict like missile systems, air traffic control systems, robots etc.

**Two types of Real-Time Operating System which are as follows:**

* **Hard Real-Time Systems:**  
  These OSs are meant for the 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 air bags which are required to be readily available in case of any accident. Virtual memory is almost never found in these systems.
* **Soft Real-Time Systems:**  
  These OSs are for applications where for time-constraint is less strict.

**Advantages of RTOS:**

* **Maximum Consumption:** Maximum utilization of devices and system,thus more output from all the resources
* **Task Shifting:** Time assigned for shifting tasks in these systems are very less. For example in older systems it takes about 10 micro seconds in shifting one task to another and in latest systems it takes 3 micro seconds.
* **Focus on Application:** Focus on running applications and less importance to applications which are in queue.
* **Real time operating system in embedded system:** Since size of programs are 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 type of systems.

**Disadvantages of RTOS:**

* **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on 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 interrupt signals to response earliest to interrupts.
* **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

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

# Difference between multitasking, multithreading and multiprocessing

1. **Multiprogramming –** A computer running more than one program at a time (like running Excel and Firefox simultaneously).
2. **Multiprocessing –** A computer using more than one CPU at a time.
3. **Multitasking –** Tasks sharing a common resource (like 1 CPU).
4. **Multithreading** is an extension of multitasking.

### ****1. Multi programming –****

In a modern computing system, there are usually several concurrent application

processes which want to execute. Now it is the responsibility of the Operating System to manage all the processes effectively and efficiently.  
One of the most important aspects of an Operating System is to multi program.  
In a computer system, there are multiple processes waiting to be executed, i.e. they are waiting when the CPU will be allocated to them and they begin their execution. These processes are also known as jobs. Now the main memory is too small to accommodate all of these processes or jobs into it. Thus, these processes are initially kept in an area called job pool. This job pool consists of all those processes awaiting allocation of main memory and CPU.  
CPU selects one job out of all these waiting jobs, brings it from the job pool to main memory and starts executing it. The processor executes one job until it is interrupted by some external factor or it goes for an I/O task.

**Non-multi programmed system’s working –**

* In a non multi programmed system, As soon as one job leaves the CPU and goes for some other task (say I/O ), the CPU becomes idle. The CPU keeps waiting and waiting until this job (which was executing earlier) comes back and resumes its execution with the CPU. So CPU remains free for all this while.
* Now it has a drawback that the CPU remains idle for a very long period of time. Also, other jobs which are waiting to be executed might not get a chance to execute because the CPU is still allocated to the earlier job.  
  This poses a very serious problem that even though other jobs are ready to execute, CPU is not allocated to them as the CPU is allocated to a job which is not even utilizing it (as it is busy in I/O tasks).
* It cannot happen that one job is using the CPU for say 1 hour while the others have been waiting in the queue for 5 hours. To avoid situations like this and come up with efficient utilization of CPU, the concept of multi programming came up.

The main idea of multi programming is to maximize the CPU time.