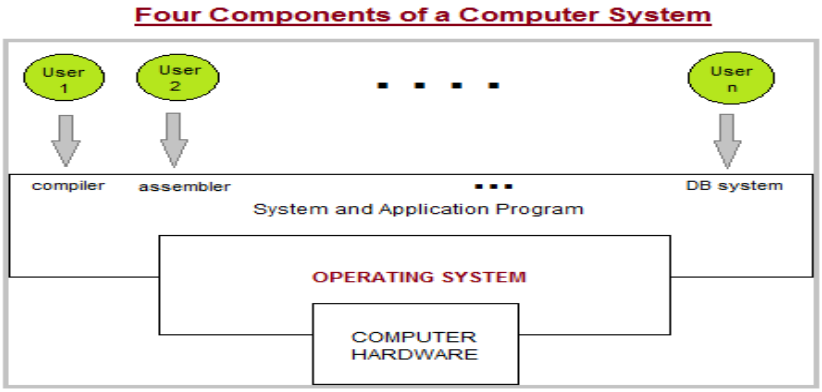
# Introduction To Operating Systems

A computer system has many resources (hardware and software), which may be required to complete a task. The commonly required resources are input/output devices, memory, file storage space, CPU, etc. The operating system acts as a manager of the above resources and allocates them to specific programs and users, whenever necessary to perform a particular task. Therefore the operating system is the resource manager i.e. it can manage the resource of a computer system internally. The resources are processor, memory, files, and I/O devices. In simple terms, an operating system is an interface between the computer user and the machine

An operating system acts similarly like government means an operating system performs no useful function by itself; though it provides an environment within which other programs can do useful work.

## **OS** is mainly designed in order to serve **two basic purposes**:

1. The operating system mainly controls the allocation and use of the computing System's resources among the various user and tasks.
2. It mainly provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding, creation of application programs and debugging



A computer system consists of many resources like - hardware and software - that must be managed efficiently. The operating system acts as the manager of the resources, decides between conflicting requests, controls the execution of programs, etc.

## Operating System Management Tasks

1. Process management which involves putting the tasks into order and pairing them into manageable size before they go to the CPU.
2. Memory management which coordinates data to and from RAM (random-access memory) and determines the necessity for virtual memory.
3. Device management provides an interface between connected devices.
4. Storage management which directs permanent data storage.
5. An application that allows standard communication between software and your computer.
6. The user interface allows you to communicate with your computer.

## Functions of Operating System

1. It boots the computer
2. It performs basic computer tasks e.g. managing the various peripheral devices e.g. mouse, keyboard
3. It provides a user interface, e.g. command line, graphical user interface (GUI)

4. It handles system resources such as the computer's memory and sharing of the central processing unit(CPU) time by various applications or peripheral devices.
5. It provides file management which refers to the way that the operating system manipulates, stores, retrieves, and saves data.
6. Error Handling is done by the operating system. It takes preventive measures whenever required to avoid errors.

**Advantages of Operating System**

1. The operating system helps to improve the efficiency of the work and helps to save a lot of time by reducing the complexity.
2. With the help of an Operating system, sharing data becomes easier with a large number of users.
3. An operating system is mainly used to hide the complexity of the hardware.
4. The different components of a system are independent of each other, thus failure of one component does not affect the functioning of another.
5. An operating system can be refreshed easily from time to time without having any problems.

**Disadvantages of Using Operating System**

1. **Expensive:** There are some open-source platforms like Linux. But some operating systems are expensive. Also, users can use free operating systems but generally, there is a bit difficulty to run them than others. On the other hand, operating systems like Microsoft Windows having GUI functionality and other in-built features are very expensive.
2. **Virus:**Threat Operating Systems are open to virus attacks and sometimes it happens that many users download the malicious

software packages on their system which pauses the functioning of the Operating system and also slows it down.
3. **Complexity:** Some operating systems are complex in nature because the language used to establish them is not clear and well defined. If there occurs an issue in the operating system then the user becomes unable to resolve that issue.
4. **System Failure:** An operating system is the heart of the computer system if due to any reason it will stop functioning then the whole system will crashes.

## Types of Operating System

### Simple Batch Systems

- In this type of system, there is no direct interaction between user and the computer.
- The user has to submit a job (written on cards or tape) to a computer operator.
- Then computer operator places a batch of several jobs on an input device.
- Jobs are batched together by type of languages and requirement.
- Then a special program, the monitor, manages the execution of each program in the batch.
- The monitor is always in the main memory and available for execution.

## Disadvantages:

The problems with Batch Systems are as follows −

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

### Time Sharing Operating System

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

**Advantages of Timesharing operating systems are as follows −**

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

**Disadvantages of Time-sharing operating systems are as follows −**

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

## 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.

**Multi programmed system's working –**

- In a multi-programmed system, *as soon as one job goes for an I/O task, the Operating System interrupts that job, chooses another job from the job pool (waiting queue), gives CPU to this new job and starts its execution.* The previous job keeps doing its I/O operation while this new job does CPU bound tasks. Now say the second job also goes for an I/O task, the CPU chooses a third job and starts executing it. As soon as a job completes its I/O operation and comes back for CPU tasks, the CPU is allocated to it.
- In this way, no CPU time is wasted by the system waiting for the I/O task to be completed.
  Therefore, the ultimate goal of multi programming is to keep the CPU busy as long as there are processes ready to execute. This way, multiple programs can be executed on a single processor by executing a part of a program at one time, a part of another program after this, then a part of another program and so on, hence executing multiple programs. Hence, the CPU never remains idle.

In the image below, program A runs for some time and then goes to waiting state. In the mean time program B begins its execution. So the CPU does not waste its resources and gives program B an opportunity to run.


## Real Time Operating Systems

Real-time operating systems (**RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines.

With an RTOS, the processing time is measured in tenths of seconds. This system is time-bound and has a fixed deadline. The processing in this type of system must occur within the specified constraints. Otherwise, This will lead to system failure.

The Real Tiem Operating System can be of three types:

**1.Hard** Real-Time **operating system:**

These operating systems guarantee that critical tasks be completed within a range of time.

For example, a robot is hired to weld a car body. If the robot welds too early or too late, the car cannot be sold, so it is a hard real-time system that requires complete car welding by robot hardly on the time.

**2.Soft real time operating system**

This operating system provides some relaxation in the time limit.

For example – Multimedia systems, digital audio systems etc. Explicit, programmer-defined and controlled processes are encountered in real-time systems.

Multitasking operation is accomplished by scheduling processes for execution independently of each other. Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services. The processor is allocated to the highest priority processes. This type of schedule, called, priority-based preemptive scheduling is used by real-time systems.

**3. Firm Real-time Operating System:**

RTOS of this type have to follow deadlines as well. In spite of its small impact, missing a deadline can have unintended consequences, including a reduction in the quality of the product. Example: Multimedia applications.

**Advantages:**

The advantages of real-time operating systems are as follows-

1. **Maximum consumption –**
   Maximum utilization of devices and systems. Thus more output from all the resources.

2. **Task Shifting –**
   Time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds. Shifting one task to another and in the latest systems, it takes 3 microseconds.

3. **Focus On Application –**
   Focus on running applications and less importance to applications that are in the queue.

4. Real-Time **Operating System In Embedded System –**
   Since the size of programs is small, RTOS can also be embedded

systems like in transport and others.

5. **Error Free –**
   These types of systems are error-free.

6. **Memory Allocation –**
   Memory allocation is best managed in these types of systems.

**Disadvantages:**

The disadvantages of real-time operating systems are as follows-

1. **Limited Tasks –**
   Very few tasks run simultaneously, and their concentration is very less on few applications to avoid errors.

2. **Use Heavy System Resources –**
   Sometimes the system resources are not so good and they are expensive as well.

3. **Complex Algorithms –**
   The algorithms are very complex and difficult for the designer to write on.

4. **Device Driver And Interrupt signals –**
   It needs specific device drivers and interrupts signals to respond earliest to interrupts.

5. **Thread Priority –**

   It is not good to set thread priority as these systems are very less prone to switching tasks.

6. **Minimum Switching –** RTOS performs minimal task switching.