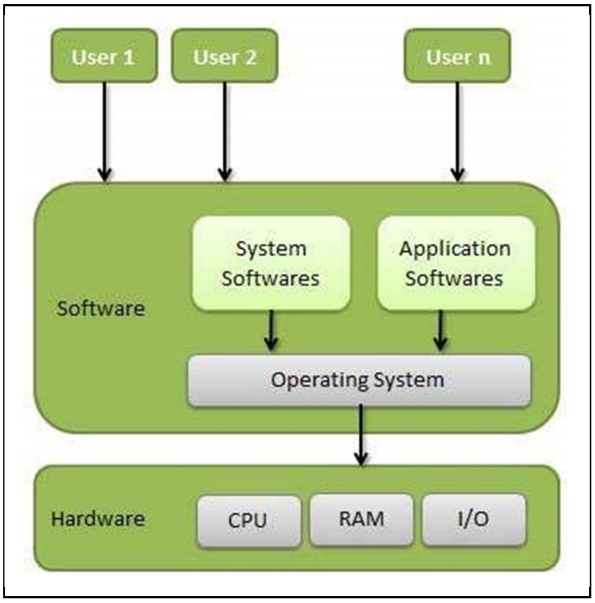
## Introduction to Operating System

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.




## Functions of Operating System

Following are some of important functions of an operating System.

1. **Memory Management:** Memory management refers to management of Primary Memory or 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.
   - Allocates the memory when a process requests it to do so.
   - De-allocates the memory when a process no longer needs it or has been terminated.
2. **Processor Management:** 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.

3. **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.
4. **File Management:** 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.
   - De-allocates the resources.

**Other Important Activities**

- **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 software and users:** Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

**Characteristics of Operating System**

The Operating System has the Following Characteristics:-

1) Operating System is a Collection of Programs those are Responsible for the Execution of other Programs.

2) Operating System is that which Responsible is for Controlling all the Input and Output Devices those are connected to the System.

3) Operating System is that which Responsible is for Running all the Application Software's.

4) Operating System is that which Provides Scheduling to the Various Processes Means Allocates the Memory to various Process those Wants to Execute.

5) Operating System is that which provides the Communication between the user and the System.

**Types of Operating Systems**

Following are some of the most widely used types of Operating system:

1) **Serial Processing:** The Serial Processing Operating Systems are those which Performs all the instructions into a Sequence Manner or the Instructions those are given by the user will be executed by using the FIFO Manner means First in First Out. All the Instructions those are Entered First in the System will be Executed First and the Instructions those are Entered Later Will be Executed Later.
2) **Batch Processing:** The Batch Processing is same as the Serial Processing Technique. But in the Batch Processing Similar Types of jobs are Firstly Prepared and they are Stored on the Card. and that card will be Submit to the System for the Processing. The System then Perform all the Operations on the Instructions one by one.
3) **Multi-Programming:** With the help of Multi-Programming we can Execute Many Programs on the System and when we are working with the Program then we can also Submit the Second or Another Program for Running and the CPU will then Execute the Second Program after the completion of the First Program.
4) **Multi-Tasking Systems:** Multi-Tasking Systems are those which can carry out several tasks simultaneously, **Example:** printing a document while carrying out calculations within a spreadsheet. Multi-tasking systems can be single user or multi-user and they can use either pre-emptive multi-tasking, where the OS handles the switching of resources between tasks, or co-operative multi-tasking.
5) **Time-sharing operating systems:** 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.
6) **Real Time Operating System:** In this Response Time is already fixed. Means time to Display the Results after Possessing has fixed by the Processor or CPU. Real Time System is used at those Places in which we Requires higher and Timely Response. These Types of Systems are used in Reservation. So when we specify the Request, the CPU will perform at that Time. There are two Types of Real Time System
    - **Hard Real Time System:** In the Hard Real Time System, Time is fixed and we can't Change any Moments of the Time of Processing. Means CPU will Process the data as we Enters the Data.

- **Soft Real Time System:** In the Soft Real Time System, some Moments can be Change. It means after giving the Command to the CPU, CPU Performs the Operation after a Microsecond.

## System Call

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 calls are used for hardware services, to create or execute a process, and for communicating with kernel services, including application and process scheduling.

## Types of System Calls

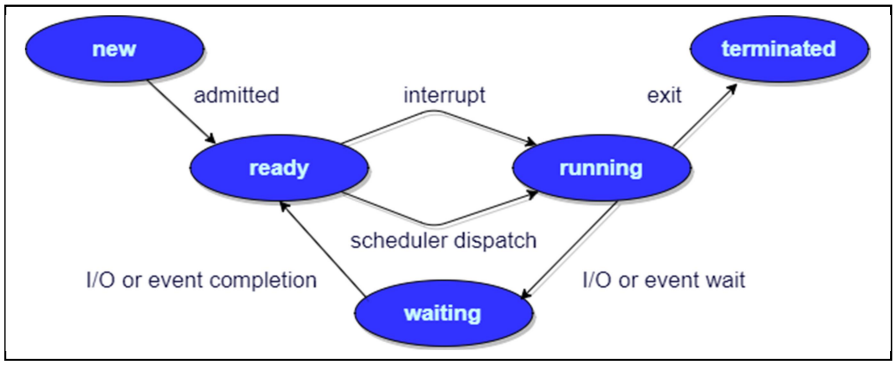
1. **Process control**
   - end, abort
   - load, execute
   - create process, terminate process
   - get process attributes, set process attributes
   - wait for time
   - wait event, signal event
   - allocate and free memory
2. **File management**
   - create file, delete file
   - open, close file
   - read, write, reposition
   - get and set file attributes
3. **Device management**
   - request device, release device
   - read, write, reposition
   - get device attributes, set device attributes
   - logically attach or detach devices
4. **Information maintenance**
   - get time or date, set time or date
   - get system data, set system data
   - get and set process, file, or device attributes
5. **Communications**
   - create, delete communication connection
   - send, receive messages
   - transfer status information
   - attach and detach remote devices

## Process

A process is basically a program in execution. The execution of a process must progress in a sequential fashion. Process is not as same as program code but a lot more than it. A process is an 'active' entity as opposed to program which is considered to be a 'passive' entity. Attributes held by process include hardware state, memory, CPU etc.

## Process Life Cycle

When a process executes, it passes through different states. Processes can be any of the following states:




**New -** The process is being created.

**Ready -** The process is waiting to be assigned to a processor.

**Running -** Instructions are being executed.

**Waiting -** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).

**Terminated -** The process has finished execution.

## Process Scheduling

The act of determining which process in the ready state should be moved to the running state is known as Process Scheduling.

The prime aim of the process scheduling system is to keep the CPU busy all the time and to deliver minimum response time for all programs. For achieving this, the scheduler must apply appropriate rules for swapping processes IN and OUT of CPU.

## Process Scheduling Queues

1. **Job queue:** This is also known as job pool. When process comes into the system and captures all other resources except memory, then they are put in job queue.
2. **Ready queue:** This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
3. **Device queue:** The processes which are blocked due to unavailability of an I/O device constitute this queue. There is a separate queue for each I/O device.

## Schedulers

Schedulers are special system software which handles process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. Schedulers are of three types −

1. **Long-Term Scheduler:** It is also called a **job scheduler**. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.
2. **Short-Term Scheduler:** It is also called as **CPU scheduler**. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects a process among the processes that are ready to execute and allocates CPU to one of them.
   Short-term schedulers, also known as **dispatchers**, make the decision of which process to execute next. Short-term schedulers are faster than long-term schedulers.
3. **Medium-Term Scheduler:** Medium-term scheduling is a part of **swapping**. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

## Scheduling Criteria

Scheduling criteria is also called as scheduling methodology. Key to multiprogramming is scheduling. There are many different criteria to check when considering the "best" scheduling algorithm:

- **CPU Utilization:** Keep the CPU as busy as possible. It ranges from 0 to 100%. In practice, it ranges from 40 to 90%.
- **Throughput:** Throughput is the rate at which processes are completed per unit of time.

- **Turnaround time:** This is the how long a process takes to execute a process. It is calculated as the time gap between the submission of a process and its completion.
- **Waiting time:** Waiting time is the sum of the time periods spent in waiting in the ready queue.
- **Load average:** It is the average number of processes residing in the ready queue waiting for their turn to get into the CPU.
- **Response time:** Response time is the time it takes to start responding from submission time. It is calculated as the amount of time it takes from when a request was submitted until the first response is produced.
- **Fairness:** Each process should have a fair share of CPU.

## Scheduling Algorithms

Scheduling algorithms or scheduling policies are mainly used for short-term scheduling. Scheduling algorithms decide which of the processes in the ready queue is to be allocated to the CPU is basis on the type of scheduling policy and whether that policy is either preemptive or non-preemptive.

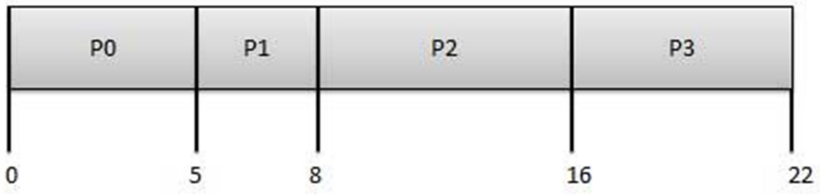
**Preemptive scheduling:** The preemptive scheduling is prioritized. The highest priority process should always be the process that is currently utilized.

**Non-Preemptive scheduling:** When a process enters the state of running, the state of that process is not deleted from the scheduler until it finishes its service time.

**First Come First Serve (FCFS) Scheduling**

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement.
- Poor in performance as average wait time is high.

| Process | Arrival Time | Execute Time | Service Time |
|---|---|---|---|
| P0 | 0 | 5 | 0 |
| P1 | 1 | 3 | 5 |
| P2 | 2 | 8 | 8 |
| P3 | 3 | 6 | 16 |




Wait time of each process is as follows –

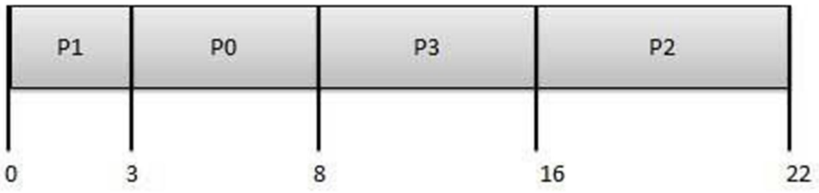
| Process | Wait Time : Service Time - Arrival Time |
|---|---|
| P0 | 0 - 0 = 0 |
| P1 | 5 - 1 = 4 |
| P2 | 8 - 2 = 6 |
| P3 | 16 - 3 = 13 |

Average Wait Time: (0+4+6+13) / 4 = 5.75

**Shortest Job Next (SJN)**

- This is also known as **shortest job first**, or **SJF**
- Best approach to minimize waiting time.
- Actual time taken by the process is already known to processor.
- Impossible to implement.

| Process | Arrival Time | Execute Time | Service Time |
|---|---|---|---|
| P0 | 0 | 5 | 3 |
| P1 | 1 | 3 | 0 |
| P2 | 2 | 8 | 16 |
| P3 | 3 | 6 | 8 |




Wait time of each process is as follows −

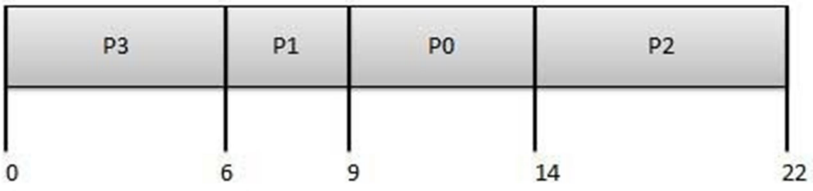
| Process | Wait Time : Service Time - Arrival Time |
|---|---|
| P0 | 3 - 0 = 3 |
| P1 | 0 - 0 = 0 |
| P2 | 16 - 2 = 14 |
| P3 | 8 - 3 = 5 |

Average Wait Time: (3+0+14+5) / 4 = 5.50

**Priority Scheduling**

- Priority is assigned for each process.
- Process with highest priority is executed first and so on.
- Processes with same priority are executed in FCFS manner.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

| Process | Arrival Time | Execute Time | Priority | Service Time |
|---|---|---|---|---|
| P0 | 0 | 5 | 1 | 9 |
| P1 | 1 | 3 | 2 | 6 |
| P2 | 2 | 8 | 1 | 14 |
| P3 | 3 | 6 | 3 | 0 |



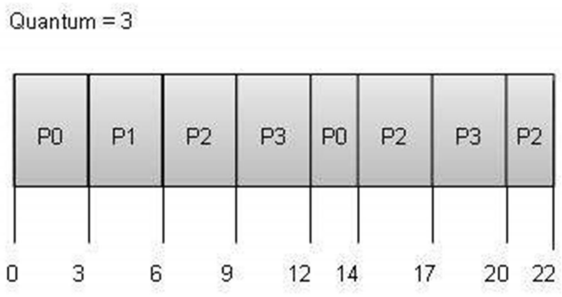

Wait time of each process is as follows −

| Process | Wait Time : Service Time - Arrival Time |
|---|---|
| P0 | 9 - 0 = 9 |
| P1 | 6 - 1 = 5 |
| P2 | 14 - 2 = 12 |
| P3 | 0 - 0 = 0 |

Average Wait Time: (9+5+12+0) / 4 = 6.5

**Round Robin(RR) Scheduling**

- A fixed time is allotted to each process, called quantum, for execution.
- Once a process is executed for given time period that process is preempted and other process executes for given time period.
- Context switching is used to save states of preempted processes.




Wait time of each process is as follows −

| Process | Wait Time : Service Time - Arrival Time |
|---|---|
| P0 | (0 - 0) + (12 - 3) = 9 |
| P1 | (3 - 1) = 2 |
| P2 | (6 - 2) + (14 - 9) + (20 - 17) = 12 |
| P3 | (9 - 3) + (17 - 12) = 11 |

Average Wait Time: (9+2+12+11) / 4 = 8.5

**Multilevel Queue Scheduling:** According to the priority of process, processes are placed in the different queues. Generally high priority processes are placed in the top level queue. Only after completion of processes from top level queue, lower level queued processes are scheduled.

**Multi-level Feedback Queue Scheduling:** It allows the process to move in between queues. The idea is to separate processes according to the characteristics of their CPU bursts. If a process uses too much CPU time, it is moved to a lower-priority queue.