UNIT - I

❖ What is Operating System?

An operating System (OS) is an intermediary between users and computer hardware. It provides users an environment in which a user can execute programs conveniently and efficiently.

The OS coordinates the use of the hardware and application programs for various users. It provides a platform for other application programs to work. The operating system is a set of special programs that run on a computer system that allows it to work properly. It controls input-output devices, execution of programs, managing files, etc.

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.

❖ Types of Operating System(Examples of OS)

- ✓ Batch Operating System (e.g., Early mainframe systems - IBM 1401)
- ✓ Time-Sharing Operating System (e.g., UNIX, Multics.)
- ✓ Real-Time Operating System (RTOS) (e.g., VxWorks, FreeRTOS.)
- ✓ Network Operating System (NOS) (e.g., Microsoft Windows Server)
- ✓ Single-User, Single-Tasking Operating System (e.g., MS-DOS)
- ✓ Single-User, Multi-Tasking Operating System (e.g., Windows, macOS)
- ✓ Multi-User Operating System (e.g., UNIX, Linux)
- ✓ Mobile Operating System (e.g., Android, iOS, HarmonyOS)
- ✓ Cloud Operating System (e.g., Google Chrome OS (for cloud-based computing), OpenStack.)

❖ Services of Operating System (Functions of OS)

- Program execution
- Input Output Operations
- Communication between Process
- File Management
- Memory Management
- Process Management
- Security and Privacy
- Resource Management
- User Interface
- Networking
- Error handling
- Time Management

✓ Program Execution

It is the Operating System that manages how a program is going to be executed. It loads the program into the memory after which it is executed. The order in which they are executed depends on the CPU Scheduling Algorithms. A few are FCFS, SJF, etc.

✓ Input Output Operations

Operating System manages the input-output operations and establishes communication between the user and device drivers. Device drivers are software that is associated with hardware that is being managed by the OS so that the sync between the devices works properly. It also provides access to input-output devices to a program when needed.

✓ Communication Between Processes

The Operating system manages the communication between processes. Communication between processes includes data transfer among them

✓ File Management

The operating system helps in managing files also. If a program needs access to a file, it is the operating system that grants access. These permissions include read-only, read-write, etc. It also provides a platform for the user to create, and delete files.

The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.

✓ 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 access directly by the CPU. So for a program to be executed, it must in the main memory. Operating System does the following activities for memory management.

✓ Process Management

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

✓ Security and Privacy

- **Security:** OS keep our computer safe from an unauthorized user by adding security layer to it. Basically, Security is nothing but just a layer of protection which protect computer from bad guys like viruses and hackers. OS provide us defenses like firewalls and anti-virus software and ensure good safety of computer and personal information.

- **Privacy:** OS give us facility to keep our essential information hidden like having a lock on our door, where only you can enter and other are not allowed. Basically, it respect our secrets and provide us facility to keep it safe.

- ✓ Resource Management

System resources are shared between various processes. It is the Operating system that manages resource sharing. It also manages the CPU time among processes using CPU Scheduling Algorithms. It also helps in the memory management of the system. It also controls input-output devices. The OS also ensures the proper use of all the resources available by deciding which resource to be used by whom.

- ✓ User Interface

User interface is essential and all operating systems provide it. Users either interacts with the operating system through the command-line interface or graphical user interface or GUI. The command interpreter executes the next user-specified command.

- ✓ Error Handling

The Operating System also handles the error occurring in the CPU, in Input-Output devices, etc. It also ensures that an error does not occur frequently and fixes the errors.

- ✓ Time Management

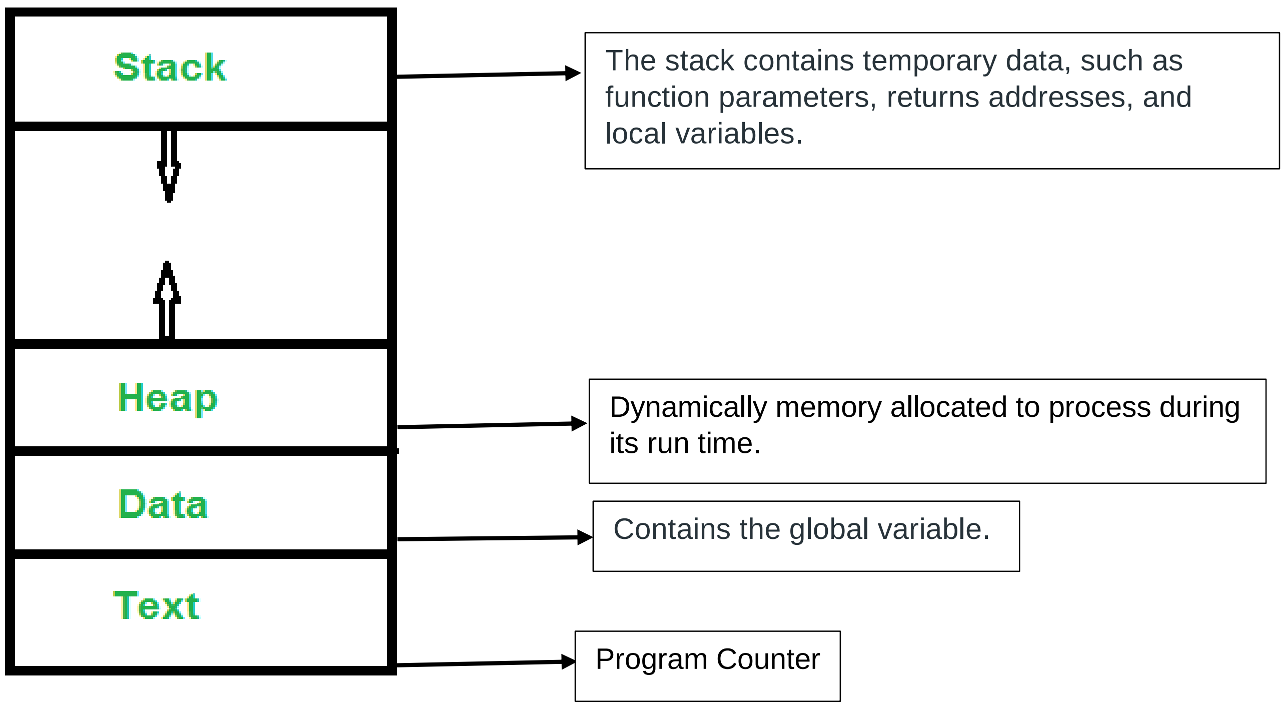
Imagine traffic light as (OS), which indicates all the cars(programs) whether it should be stop(red)=>(simple queue), start(yellow)=>(ready queue),move(green)=>(under execution) and this light (control) changes after a certain interval of time at each side of the road(computer system) so that the cars(program) from all side of road move smoothly without traffic.

❖ Introduction Process Management

Process management is a key part of an operating system. It controls how processes are carried out, and controls how your computer runs by handling the active processes. This includes stopping processes, setting which processes should get more attention, and many more.

The OS is responsible for managing the start, stop, and scheduling of processes, which are programs running on the system.

A process in memory is divided into several distinct sections, each serving a different purpose. Here's how a process typically looks in memory:

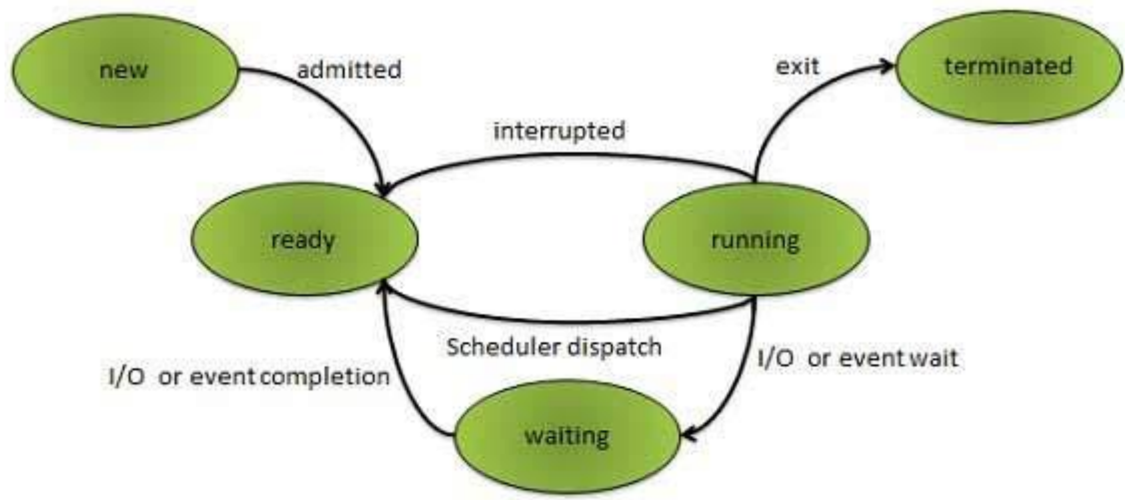


✓ Characteristics of a Process

- Process Id: A unique identifier assigned by the operating system.
- Process State: Can be ready, running, etc.
- CPU Registers: Like the Program Counter (CPU registers must be saved and restored when a process is swapped in and out of the CPU)
- Accounts Information: Amount of CPU used for process execution, time limits, execution ID, etc
- I/O Status Information: For example, devices allocated to the process, open files, etc
- CPU Scheduling Information: For example, Priority (Different processes may have different priorities)

❖ Process States

As a process executes, it changes state. The state of a process is defined as the current activity of the process.

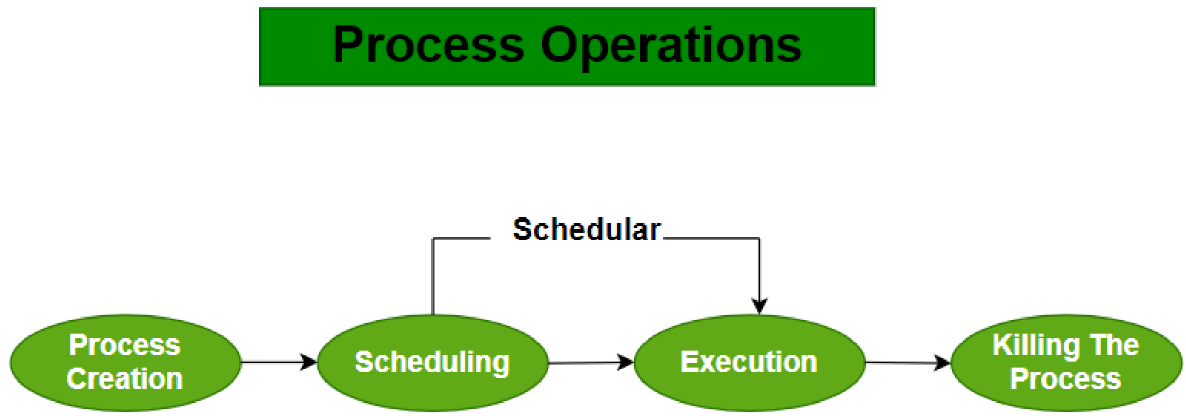


Process can have one of the following five states at a time.

1. **New:** The process is being created.
2. **Ready:** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run.
3. **Running**: Process instructions are being executed (i.e. The process that is currently being executed).
4. **Waiting**: The process is waiting for some event to occur (such as the completion of an I/O operation).
5. **Terminated**: The process has finished execution.

✓ Process Operations

Process operations in an operating system refer to the various activities the OS performs to manage processes. These operations include process creation, process scheduling, execution and killing the process. Here are the key process operations:



1. Creation

Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.

2. Scheduling

Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.

3. Execution

Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.

4. Deletion/killing

Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.

❖ Process Control Block (PCB)

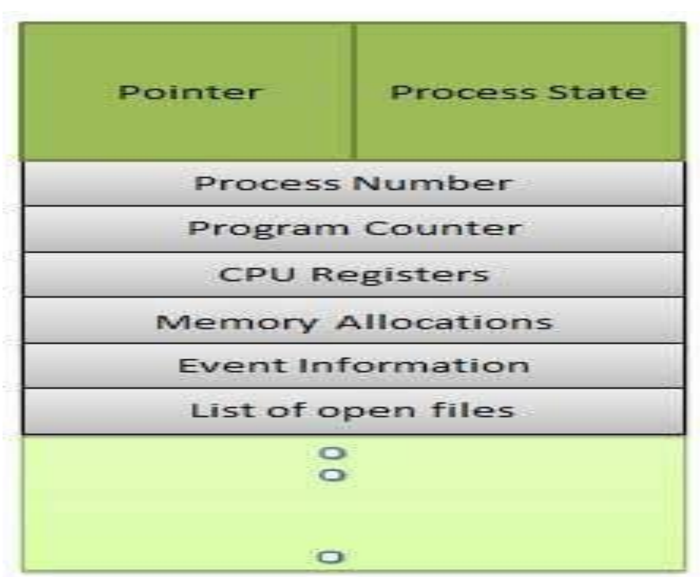
The operating system uses a data structure called a Process Control Block (PCB) to store and handle process-related data.

Each process is represented in the operating system by a process control block (PCB) also called a task control block. PCB is the data structure used by the operating system.

A Process Control Block (PCB) is a data structure used by the operating system to manage information about a process. The process control keeps track of many important pieces of information needed to manage processes efficiently.

Real-time operating systems may require additional information in the PCB, such as deadlines and priorities, to ensure that time-critical processes are executed in a timely manner. The Process Control Block (PCB) is stored in a special part of memory that normal users can't access.

The process of switching from one process to another is called context switching. The PCB plays a crucial role in context switching by saving the state of the current process and restoring the state of the next process.



1. **Pointer**: Pointer is a stack pointer which is required to be saved when the process is Switched from one state to another state to retain the current position of the process

2. **Process State**: Process state may be new, ready, running, waiting and so on.

3. **Program Counter:** Program Counter indicates the address of the next instruction to be executed for this process.

4. **CPU registers**: CPU registers include general purpose register, base register accumulators etc. number of register and type of register totally depends upon the computer architecture.

5. **Memory management information:** This information may include the value of base and limit registers, the page tables, or the segment tables depending on the memory system used by the operating system. This information is useful for deallocating the memory when the process terminates.

6. **Accounting/Event information:** This information includes the amount of CPU time used, time limits, job or process numbers, account numbers etc.

❖ Advantages

- **Efficient Process Management:** The PCB provide an efficient way to manage processes in an operating system. The process table contains all the information about each process, while the PCB contains the current state of the process, such as the program counter and CPU registers.

- **Resource Management:** The PCB allow the operating system to manage system resources, such as memory and CPU time, efficiently. By keeping track of each process's resource usage, the operating system can ensure that all processes have access to the resources they need.

- **Process Synchronization:** The PCB can be used to synchronize processes in an operating system. The PCB contains information about each process's synchronization state, such as its waiting status and the resources it is waiting for.

- **Process Scheduling:** The PCB can be used to schedule processes for execution. By keeping track of each process's state and resource usage, the operating system can determine which processes should be executed next.

❖ Disadvantages

- **Overhead:** The PCB can introduce overhead and reduce system performance. The operating system must maintain the process table and PCB for each process, which can consume system resources.

- **Complexity:** The PCB can increase system complexity and make it more challenging to develop and maintain operating systems. The need to manage and synchronize multiple processes can make it more difficult to design and implement system features and ensure system stability.

- **Scalability:** The PCB may not scale well for large-scale systems with many processes. As the number of processes increases, the process table and PCB can become larger and more difficult to manage efficiently.

- **Security:** The PCB can introduce security risks if they are not implemented correctly. Malicious programs can potentially access or modify the process table and PCB to gain unauthorized access to system resources or cause system instability.

❖ Process Scheduling Queues

In an operating system (OS), process scheduling involves managing processes as they move between different states (such as ready, running, waiting, etc.) in the process life cycle. To handle this, the OS uses various queues to manage processes at different stages of their execution. Three of the most important queues involved in process scheduling are:

- ✓ **Job Queue:**

  Definition: The job queue is the queue where all processes (or jobs) are placed when they are first created or submitted to the system. It contains all the processes that are in the system, including those that have not yet been scheduled for execution.

  Role: The job queue holds the processes before they are admitted into the ready queue. Once a process is ready for execution, it moves from the job queue to the ready queue.

- ✓ **Ready Queue:**

  Definition: The ready queue is the queue where processes reside when they are ready to be executed, but the CPU is not yet allocated to them. Processes in the ready queue are waiting for the CPU to be assigned by the CPU scheduler.

  Role: This queue holds processes that are ready to run, and processes are dispatched to the CPU from here based on scheduling algorithms (like round-robin, priority scheduling, etc.).

- ✓ **Device Queue:**

  Definition: The device queue holds processes that are waiting for a particular I/O device (such as a hard disk, printer, or network interface) to become available.

  Role: The device queue manages processes waiting for I/O operations to complete. Once the required device becomes available and the I/O operation completes, the process is moved back to the ready queue to continue execution.

❖ 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

- Long Term Schedulers
- Short Term Schedulers
- Medium Term Schedulers

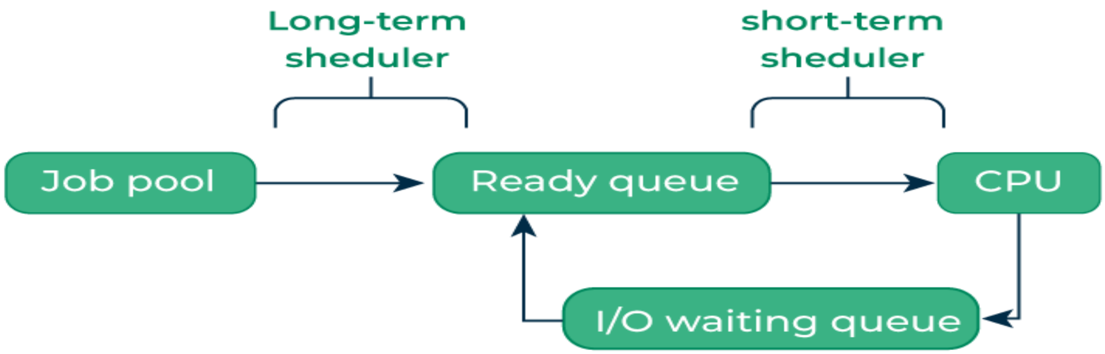
1. Long Term or Job Scheduler

It brings the new process to the ‘Ready State’. It controls the Degree of Multi-programming, i.e., the number of processes present in a ready state at any point in time. It is important that the long-term scheduler make a careful selection of both I/O and CPU-bound processes. I/O-bound tasks are which use much of their time in input and output operations while CPU-bound processes are which spend their time on the CPU. The job scheduler increases efficiency by maintaining a balance between the two. They operate at a high level and are typically used in batch-processing systems.

2. Short-Term or CPU Scheduler

It is responsible for selecting one process from the ready state for scheduling it on the running state.

Note: Short-term scheduler only selects the process to schedule it doesn’t load the process on running. Here is when all the scheduling algorithms are used. The CPU scheduler is responsible for ensuring no starvation due to high burst time processes.

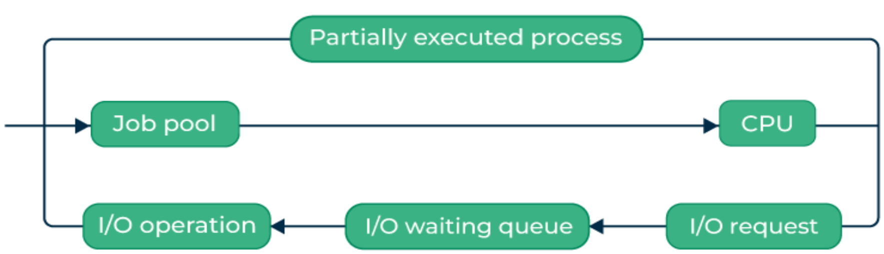


The dispatcher is responsible for loading the process selected by the Short-term scheduler on the CPU (Ready to Running State) Context switching is done by the dispatcher only. A dispatcher does the following:

- Switching context.
- Switching to user mode.
- Jumping to the proper location in the newly loaded program.

3. Medium-Term Scheduler

It is responsible for suspending and resuming the process. It mainly does swapping (moving processes from main memory to disk and vice versa). Swapping may be necessary to improve the process mix or because a change in memory requirements has overcommitted available memory, requiring memory to be freed up. It is helpful in maintaining a perfect balance between the I/O bound and the CPU bound. It reduces the degree of multiprogramming.



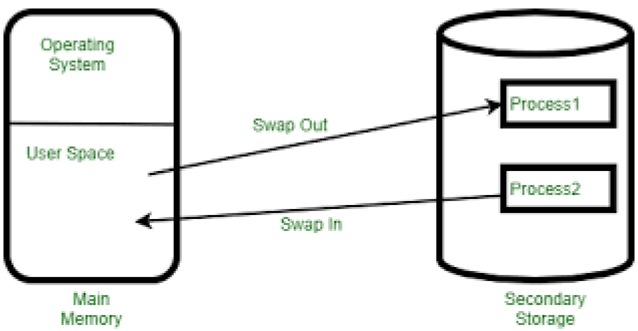
Comparison Among Scheduler

Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
It is a job scheduler	It is a CPU scheduler	It is a process-swapping scheduler.
Generally, Speed is lesser than short term scheduler	Speed is the fastest among all of them.	Speed lies in between both short and long-term schedulers.
It controls the degree of multiprogramming	It gives less control over how much multiprogramming is done.	It reduces the degree of multiprogramming.
It is barely present or nonexistent in the time-sharing system.	It is a minimal time-sharing system.	It is a component of systems for time sharing.
It can re-enter the process into memory, allowing for the continuation of execution.	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

❖ Context Switching

In order for a process execution to be continued from the same point at a later time, context switching is a mechanism to store and restore the state or context of a CPU in the Process Control block. A context switcher makes it possible for multiple processes to share a single CPU using this method. A multitasking operating system must include context switching among its features.

The state of the currently running process is saved into the process control block when the scheduler switches the CPU from executing one process to another. The state used to set the computer, registers, etc. for the process that will run next is then loaded from its own PCB. After that, the second can start processing.



Context Switching

In order for a process execution to be continued from the same point at a later time, context switching is a mechanism to store and restore the state or context of a CPU in the Process Control block.

A context switcher makes it possible for multiple processes to share a single CPU using this method. A multitasking operating system must include context switching among its features.

- Program Counter
- Scheduling information
- The base and limit register value
- Currently used register
- Changed State
- I/O State information
- Accounting information

❖ CPU Scheduling Criteria

What is CPU scheduling?

CPU Scheduling is a process that allows one process to use the CPU while another process is delayed due to unavailability of any resources such as I / O etc, thus making full use of the CPU.

CPU scheduling decides the order and priority of the processes to run and allocates the CPU time based on various parameters such as CPU usage, throughput, turnaround, waiting time, and response time.

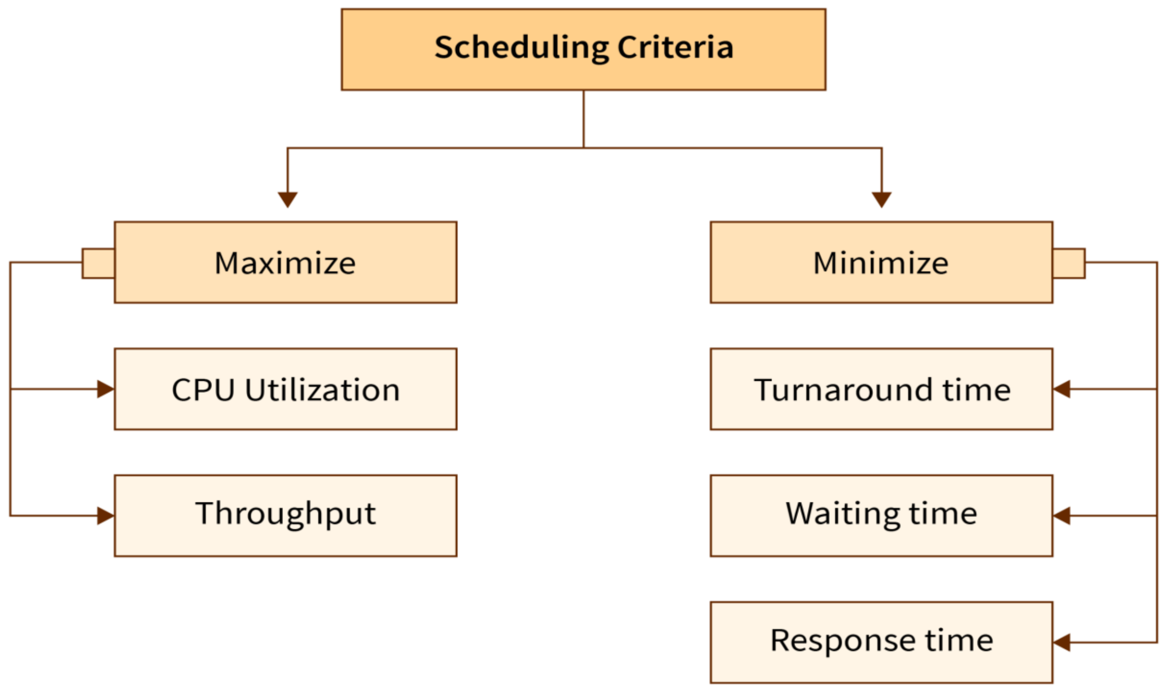
CPU scheduling is the process of determining which process or task is to be executed by the central processing unit (CPU) at any given time.

It is an important component of modern operating systems that allows multiple processes to run simultaneously on a single processor.

The purpose of CPU Scheduling is to make the system more efficient, faster, and fairer.

CPU scheduling criteria, such as turnaround time, waiting time, and throughput, are essential metrics used to evaluate the efficiency of scheduling algorithms.

Now let’s discuss CPU Scheduling has several criteria. Some of them are mentioned below.



CPU Scheduling Criteria

1. **CPU utilization**

The main objective of any CPU scheduling algorithm is to keep the CPU as busy as possible. Theoretically, CPU utilization can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the load upon the system.

2. **Throughput**

A measure of the work done by the CPU is the number of processes being executed and completed per unit of time. This is called throughput. The throughput may vary depending on the length or duration of the processes.

3. **Turnaround Time**

For a particular process, an important criterion is how long it takes to execute that process. The time elapsed from the time of submission of a process to the time of completion is known as the turnaround time. Turnaround time is the sum of times spent waiting to get into memory, waiting in the ready queue, executing in CPU, and waiting for I/O.

Turn Around Time = Completion Time – Arrival Time.

4. **Waiting Time**

A scheduling algorithm does not affect the time required to complete the process once it starts execution. It only affects the waiting time of a process i.e. time spent by a process waiting in the ready queue.

Waiting Time = Turnaround Time – Burst Time.

5. **Response Time**

In an interactive system, turn-around time is not the best criterion. A process may produce some output fairly early and continue computing new results while previous results are being output to the user. Thus another criterion is the time taken from submission of the process of the request until the first response is produced. This measure is called response time.

Response Time = CPU Allocation Time(when the CPU was allocated for the first) – Arrival Time

❖ Process Scheduling Algorithms

There are various algorithms which are used by the Operating System to schedule the processes on the processor in an efficient way.

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.

These algorithms are either **non-preemptive or preemptive**. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

Different Types of CPU Scheduling Algorithms

There are mainly two types of scheduling methods:

- **Preemptive Scheduling:** Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.

- **Non-Preemptive Scheduling:** Non-Preemptive scheduling is used when a process terminates, or when a process switches from running state to waiting state.

The Purpose of a Scheduling algorithm

- Maximum CPU utilization
- Fare allocation of CPU
- Maximum throughput
- Minimum turnaround time
- Minimum waiting time
- Minimum response time

Important Abbreviations

1. CPU - - - > Central Processing Unit
2. AT - - - > Arrival Time
3. BT - - - > Burst Time
4. WT - - - > Waiting Time
5. TAT - - - > Turn Around Time
6. CT - - - > Completion Time
7. FIFO - - - > First In First Out
8. TQ - - - > Time Quantum
9. FCFS - - - > First Come First Serve

1. First Come First Serve(FCFS)

FCFS considered to be the simplest of all operating system scheduling algorithms.

First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first and is implemented by using FIFO queue.

- Jobs are executed on first come, first serve basis.
- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high.

✓ **Characteristics of FCFS**

- FCFS supports non-preemptive and preemptive CPU scheduling algorithms.
- Tasks are always executed on a First-come, First-serve concept.
- FCFS is easy to implement and use.
- This algorithm is not much efficient in performance, and the wait time is quite high.

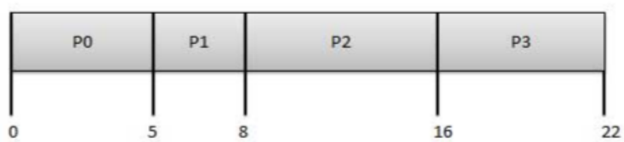
✓ **Advantages of FCFS**

- Easy to implement
- First come, first serve method

✓ **Disadvantages of FCFS**

- FCFS suffers from **Convoy effect**.
- The average waiting time is much higher than the other algorithms.
- FCFS is very simple and easy to implement and hence not much efficient.

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

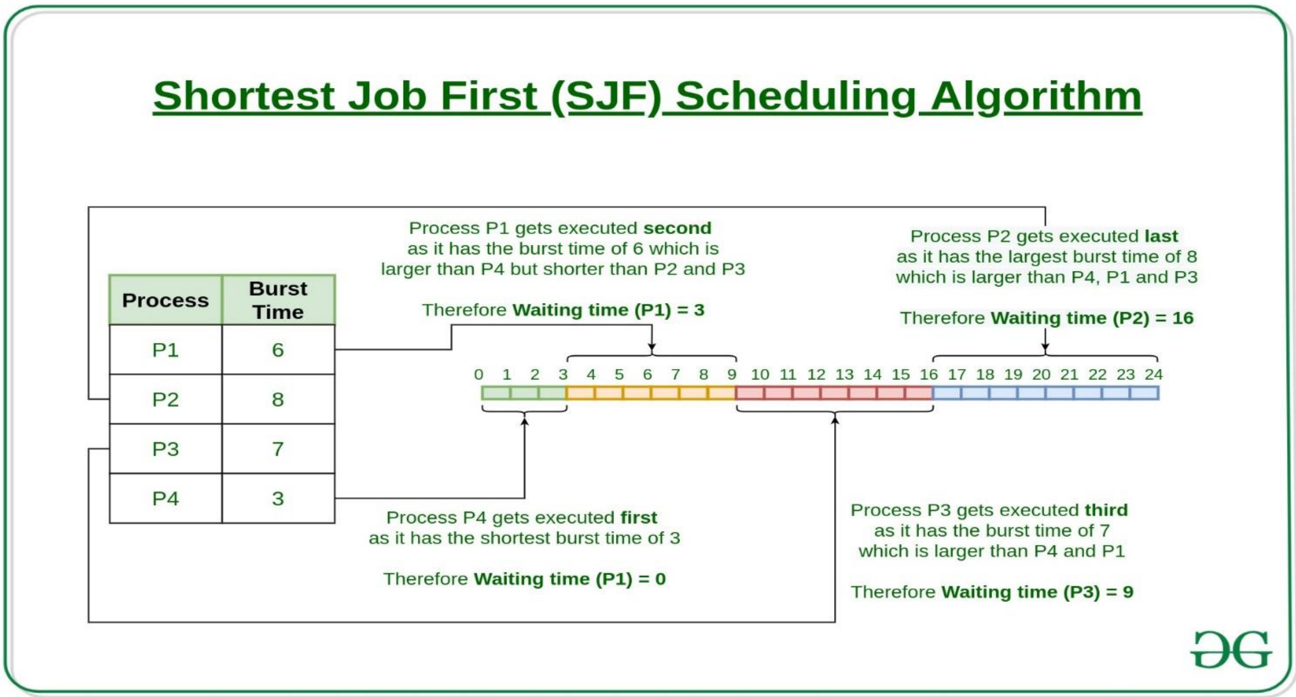
2. Shortest Job First(SJF)

Shortest job first (SJF) is a scheduling process that selects the waiting process with the smallest execution time to execute next.

This scheduling method may or may not be preemptive. Significantly reduces the average waiting time for other processes waiting to be executed. The full form of SJF is Shortest Job First.

- This is also known as **Shortest Job Next**, or SJN
- This is a non-preemptive, pre-emptive scheduling algorithm.

- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.



Given: Table of processes, and their Arrival time, Execution time

Process	Arrival Time	Execution Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	14
P3	3	6	8

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

Waiting time of each process is as follows −

Process	Waiting Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	14 - 2 = 12
P3	8 - 3 = 5

Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25

✓ Characteristics of SJF

- Shortest Job first has the advantage of having a minimum average waiting time among all operating system scheduling algorithms.
- It is associated with each task as a unit of time to complete.
- It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of ageing.

✓ Advantages of SJF

- As SJF reduces the average waiting time thus, it is better than the first come first serve scheduling algorithm.
- SJF is generally used for long term scheduling

✓ **Disadvantages of SJF**

- One of the demerit SJF has is starvation.
- Many times it becomes complicated to predict the length of the upcoming CPU request

3. Round Robin

Round Robin is a CPU scheduling algorithm where each process is cyclically assigned a fixed time slot.

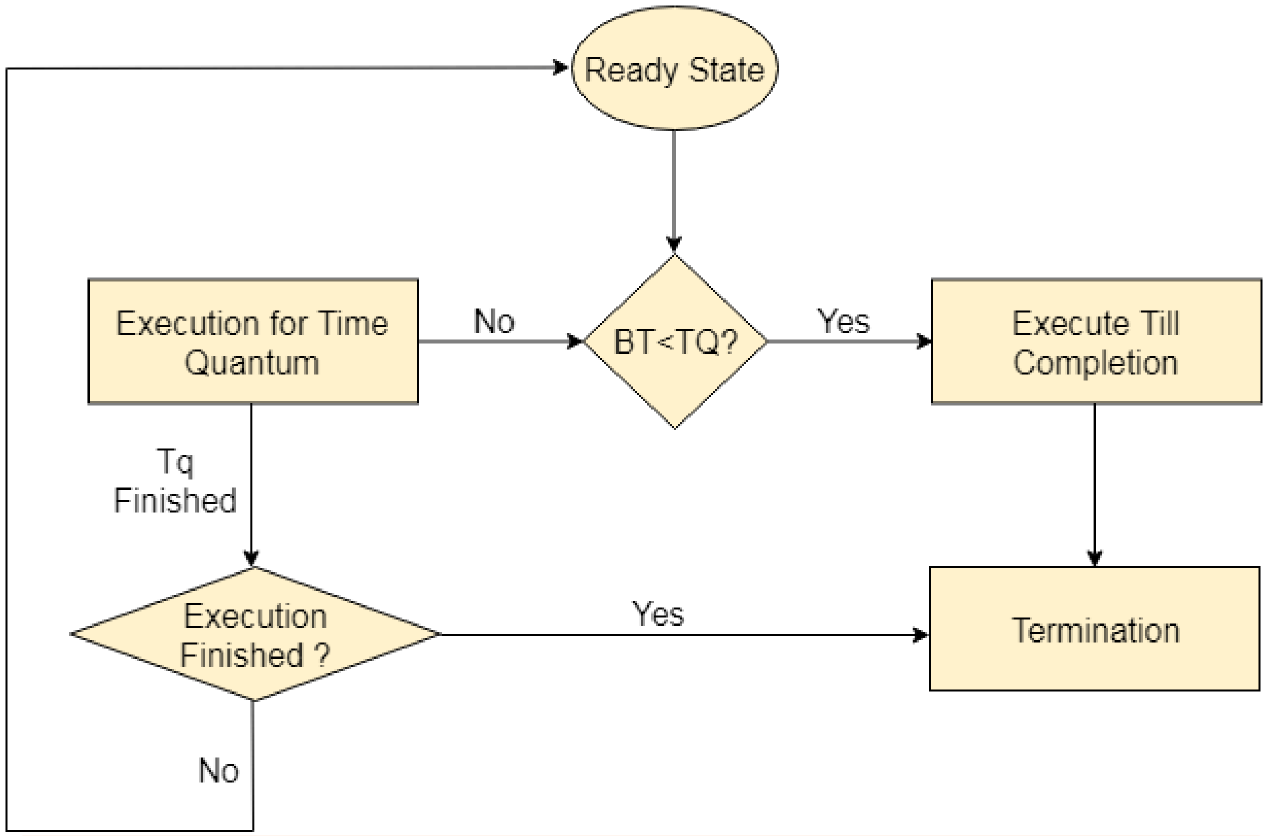
It is the preemptive version of First come First Serve CPU Scheduling algorithm. Round Robin CPU Algorithm generally focuses on Time Sharing technique.

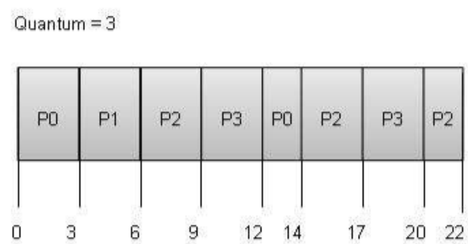
Round Robin CPU Scheduling is the most important CPU Scheduling Algorithm which is ever used in the history of CPU Scheduling Algorithms.

Round Robin CPU Scheduling uses Time Quantum (TQ).

The Time Quantum is something which is removed from the Burst Time and lets the chunk of process to be completed.

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted and other process executes for a 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

✓ **Characteristics of Round robin**

- It's simple, easy to use, and starvation-free as all processes get the balanced CPU allocation.
- One of the most widely used methods in CPU scheduling as a core.
- It is considered preemptive as the processes are given to the CPU for a very limited time.

✓ **Advantages of Round robin**

- Round robin seems to be fair as every process gets an equal share of CPU.
- The newly created process is added to the end of the ready queue.

4. Priority Scheduling

Preemptive Priority CPU Scheduling Algorithm is a pre-emptive method of CPU scheduling algorithm that works **based on the priority** of a process.

In this algorithm, the editor sets the functions to be as important, meaning that the most important process must be done first.

In the case of any conflict, that is, where there is more than one process with equal value, then the most important CPU planning algorithm works on the basis of the FCFS (First Come First Serve) algorithm.

✓ **Characteristics of Priority Scheduling**

- Schedules tasks based on priority.
- When the higher priority work arrives and a task with less priority is executing, the higher priority process will takes the place of the less priority process and
- The later is suspended until the execution is complete.
- Lower is the number assigned, higher is the priority level of a process.

✓ **Advantages of Priority Scheduling**

- The average waiting time is less than FCFS
- Less complex

✓ Disadvantages of Priority Scheduling

- One of the most common demerits of the Preemptive priority CPU scheduling algorithm is the Starvation Problem. This is the problem in which a process has to wait for a longer amount of time to get scheduled into the CPU. This condition is called the starvation problem.

Thank You