**Unit II**

**Process Scheduling**

## Definition

The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

Process scheduling is an essential part of a Multiprogramming operating systems. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.

**Categories of Scheduling**

There are two categories of scheduling:

1. **Non-preemptive:** Here the resource can’t be taken from a process until the process completes execution. The switching of resources occurs when the running process terminates and moves to a waiting state.
2. **Preemptive:** Here the OS allocates the resources to a process for a fixed amount of time. During resource allocation, the process switches from running state to ready state or from waiting state to ready state. This switching occurs as the CPU may give priority to other processes and replace the process with higher priority with the running process.

**Process Scheduling Queues**

The OS maintains all Process Control Blocks (PCBs) in Process Scheduling Queues. The OS maintains a separate queue for each of the process states and PCBs of all processes in the same execution state are placed in the same queue. When the state of a process is changed, its PCB is unlinked from its current queue and moved to its new state queue.

The Operating System maintains the following important process scheduling queues −

**Job queue −** This queue keeps all the processes in the system.

**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.

**Device queues −** The processes which are blocked due to unavailability of an I/O device constitute this queue.



The OS can use different policies to manage each queue (FIFO, Round Robin, Priority, etc.). The OS scheduler determines how to move processes between the ready and run queues which can only have one entry per processor core on the system; in the above diagram, it has been merged with the CPU.

**Schedulers**

Schedulers are special system software which handle 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 Scheduler

Short-Term Scheduler

Medium-Term Scheduler

**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.

The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

On some systems, the long-term scheduler may not be available or minimal. Time-sharing operating systems have no long term scheduler. When a process changes the state from new to ready, then there is use of long-term scheduler.

**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.

**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.

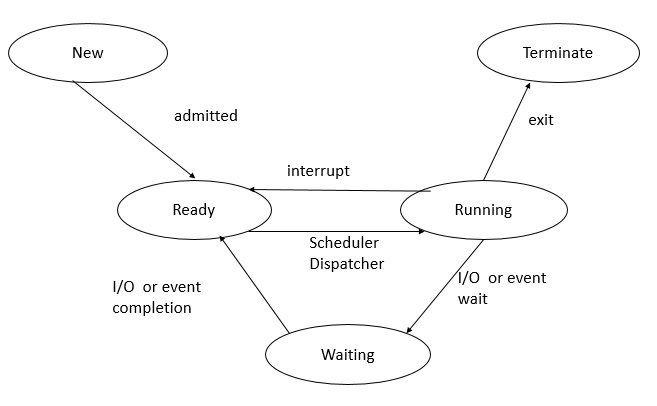
A running process may become suspended if it makes an I/O request. A suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other processes, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.

**Comparison among Scheduler**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Long-Term Scheduler | Short-Term Scheduler | Medium-Term Scheduler |
| 1 | It is a job scheduler | It is a CPU scheduler | It is a process swapping scheduler. |
| 2 | Speed is lesser than short term scheduler | Speed is fastest among other two | Speed is in between both short and long term scheduler. |
| 3 | It controls the degree of multiprogramming | It provides lesser control over degree of multiprogramming | It reduces the degree of multiprogramming. |
| 4 | It is almost absent or minimal in time sharing system | It is also minimal in time sharing system | It is a part of Time sharing systems. |
| 5 | It selects processes from pool and loads them into memory for execution | It selects those processes which are ready to execute | It can re-introduce the process into memory and execution can be continued. |

# Process States

# State Diagram



**1. New (Create)**

In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by OS to create the process.

**2. Ready**

Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory.

The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.

**3. Running**

One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running processes for a particular time will always be one. If we have n processors in the system then we can have n processes running simultaneously.

**4. Block or wait**

From the Running state, a process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behavior of the process.

When a process waits for a certain resource to be assigned or for the input from the user then the OS move this process to the block or wait state and assigns the CPU to the other processes.

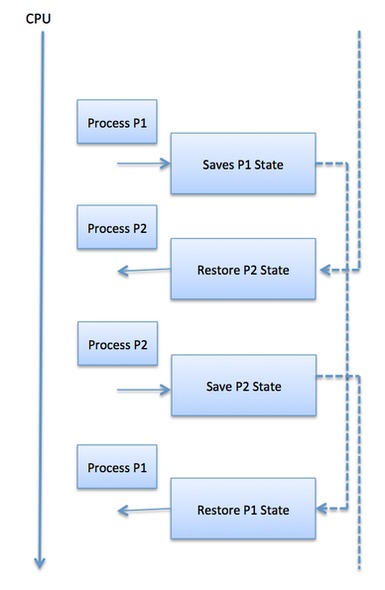
**5. Completion or termination**

When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.

**Context Switching**

A context switching is the mechanism to store and restore the state or context of a CPU in Process Control block so that a process execution can be resumed from the same point at a later time. Using this technique, a context switcher enables multiple processes to share a single CPU. Context switching is an essential part of a multitasking operating system features.

When the scheduler switches the CPU from executing one process to execute another, the state from the current running process is stored into the process control block. After this, the state for the process to run next is loaded from its own PCB and used to set the PC, registers, etc. At that point, the second process can start executing.



Context switches are computationally intensive since register and memory state must be saved and restored. To avoid the amount of context switching time, some hardware systems employ two or more sets of processor registers. When the process is switched, the following information is stored for later use.

Program Counter

Scheduling information

Base and limit register value

Currently used register

Changed State

I/O State information

Accounting information

# Scheduling algorithms

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms which we are going to discuss in this chapter −

**First-Come, First-Served (FCFS) Scheduling**

**Shortest-Job-Next (SJN) Scheduling**

**Priority Scheduling**

**Shortest Remaining Time**

**Round Robin(RR) Scheduling**

**Multiple-Level Queues Scheduling**

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.

**Arrival Time**: Time at which the process arrives in the ready queue.

**Completion Time**: Time at which process completes its execution.

**Burst Time**: Time required by a process for CPU execution.

**Turn Around Time**: Time Difference between completion time and arrival time.

**Turn Around Time** = Completion Time  –  Arrival Time

**Waiting Time(W.T):** Time Difference between turn around time and burst time.

**Waiting Time** = Turn Around Time  –  Burst Time

# First Come First Serve (FCFS)

* 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.

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on [First come, First serve Scheduling.](https://www.geeksforgeeks.org/first-come-first-serve-cpu-scheduling-non-preemptive/)

### 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.

**Characteristics of SJF:**

* Shortest Job first has the advantage of having a minimum average waiting time among all [operating system scheduling algorithms.](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/)
* 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 Shortest Job first:**

* 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

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on [Shortest Job First.](https://www.geeksforgeeks.org/program-for-shortest-job-first-or-sjf-cpu-scheduling-set-1-non-preemptive/)

### 3. Priority Scheduling:

**Preemptive Priority CPU Scheduling Algorithm** is a pre-emptive method of [CPU scheduling algorithm](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/) 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 are more than one processor 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 while a task with less priority is executed, the higher priority work takes the place of the less priority one and
* The latter 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](https://www.geeksforgeeks.org/starvation-and-aging-in-operating-systems/). 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.

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on [Priority Preemptive Scheduling algorithm](https://www.geeksforgeeks.org/preemptive-priority-cpu-scheduling-algortithm/).

### 4. Round robin:

**Round Robin** is a [CPU scheduling algorithm](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/) where each process is cyclically assigned a fixed time slot. It is the [preemptive](https://www.geeksforgeeks.org/preemptive-and-non-preemptive-scheduling/)version of[First come First Serve CPU Scheduling algorithm](https://www.geeksforgeeks.org/first-come-first-serve-cpu-scheduling-non-preemptive/). Round Robin CPU Algorithm generally focuses on Time Sharing technique.

**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.

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on the [Round robin Scheduling algorithm](https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1/).

### 5. Shortest Remaining Time First:

**Shortest remaining time first** is the preemptive version of the Shortest job first which we have discussed earlier where the processor is allocated to the job closest to completion. In SRTF the process with the smallest amount of time remaining until completion is selected to execute.

**Characteristics of** **Shortest remaining time first:**

* SRTF algorithm makes the processing of the jobs faster than SJF algorithm, given it’s overhead charges are not counted.
* The context switch is done a lot more times in SRTF than in SJF and consumes the CPU’s valuable time for processing. This adds up to its processing time and diminishes its advantage of fast processing.

**Advantages of SRTF:**

* In SRTF the short processes are handled very fast.
* The system also requires very little overhead since it only makes a decision when a process completes or a new process is added.

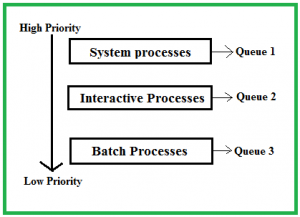
**Disadvantages of SRTF:**

* Like the shortest job first, it also has the potential for process starvation.
* Long processes may be held off indefinitely if short processes are continually added.

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on the [shortest remaining time first](https://www.geeksforgeeks.org/shortest-remaining-time-first-preemptive-sjf-scheduling-algorithm/).

### 6. Multiple Queue Scheduling:

Processes in the ready queue can be divided into different classes where each class has its own scheduling needs. For example, a common division is a **foreground (interactive)** process and a **background (batch)** process. These two classes have different scheduling needs. For this kind of situation **Multilevel Queue Scheduling** is used.



The description of the processes in the above diagram is as follows:

* **System Processes:**The CPU itself has its process to run, generally termed as System Process.
* **Interactive Processes:**An Interactive Process is a type of process in which there should be the same type of interaction.
* **Batch Processes:**Batch processing is generally a technique in the Operating system that collects the programs and data together in the form of a **batch** before the **processing** starts.

**Advantages of multilevel queue scheduling:**

* The main merit of the multilevel queue is that it has a low scheduling overhead.

**Disadvantages of multilevel queue scheduling:**

* Starvation problem
* It is inflexible in nature

To learn about how to implement this CPU scheduling algorithm, please refer to our detailed article on [Multilevel Queue Scheduling](https://www.geeksforgeeks.org/multilevel-queue-mlq-cpu-scheduling/).