

# CPU Scheduling Algorithms in Operating Systems

In this tutorial, we will be learning about the CPU Scheduling Algorithms in Operating Systems. These are algorithms are very important topic in Operating Systems. This is because this CPU Scheduling Algorithms forms a base and foundation for the Operating Systems subject.

There are many processes which are going on in the Operating System. A task is a group of processes. Every task is executed by the Operating System. The Operating System divides the task into many processes. The final goal of the Operating System is completion of the task.

But there are some certain conditions which must be followed by the task. The conditions are that the Task must be finished in the quickest possible time with the limited resources which the Operating System have. This is the main motive of CPU Scheduling Algorithms.

## CPU Scheduling

The CPU Scheduling is the process by which a process is executed by the using the resources of the CPU. The process also can wait due to the absence or unavailability of the resources. These processes make the complete use of Central Processing Unit.

The operating system must choose one of the processes in the list of ready-to-launch processes whenever the CPU gets idle. A temporary (CPU) scheduler does the selection. The Scheduler choose one of the ready-to-start memory processes to get the CPU.

Before, going to the Types of CPU Scheduling Algorithms, we are going to learn about the Basic Terminologies which are to be followed and used in the CPU Scheduling Algorithms by us.

### 1. Process ID

The Process ID is the first Thing is to be written while solving the problem. The Process ID acts like the name of the process. It is usually represented with numbers or P letter with numbers

**Example:**

1. 0, 1, 2, 3, . . . . .
2. P0, P1, P2, P3 . . . . .

Usually, we start the naming of the process from zero. We can also start the numbering from number 1 also. It is our interest

### 1. Arrival Time

The time which is required for the Process to enter the ready queue or the time when the Process is ready to be executed by the CPU. This Arrival Time can be represented as AT in short form. The Arrival Times is always positive or also zero.

## **2. Burst Time**

The Time Slot which the Process requires to complete the Process is known as the Burst Time. The Burst Time can be represented as BT in short form. The Burst Times of a process is always greater than zero.

## **3. Completion Time**

The Total Time required by the CPU to complete the process is known as Completion Time. The Completion Time can be represented as CT in short form. The Completion will always be greater than zero.

## **4. Turn Around Time**

The time taken by the CPU since the Process has been ready to execute or since the process is in Ready Queue is known as Turn Around Time. The Turn Around Time can be calculated with the help of Completion Time and Arrival Time. The Turn Around Time can be represented as TAT in short form.

The Turn Around Time is the difference of Completion Time and Arrival Time.

### **Formula**

1.  $TAT = CT - AT$

Here, CT is Completion Time and AT is Arrival Time.

## **5. Waiting Time**

The time the Process has been waiting to complete its process since the assignment of process for completion is known as Waiting Time. The Waiting Time can be represented as WT in short form. The Waiting Time can be calculated with the help of Turn Around Time and Burst Time.

The Waiting Time is the difference between Turn Around Time and Burst Time

### **Formula**

1.  $WT = TAT - BT$

## **6. Ready Queue**

The Queue where all the processes are stored until the execution of the previous process. This ready queue is very important because there would be confusion in CPU when two same kinds of processes are being executed at the same time.

Then, in these kinds of conditions the ready queue comes into place and then, the its duty is fulfilled.

## 7. Gantt Chart

It is the place where all the already executed processes are stored. This is very useful for calculating Waiting Time, Completion Time, Turn Around Time.

## Modes in CPU Scheduling Algorithms

There are two modes in CPU Scheduling Algorithms. They are:

1. Pre-emptive Approach
2. Non Pre-emptive Approach

In Pre Emptive-Approach the process once starts its execution then the CPU is allotted to the same process until the completion of process. There would be no shift of Processes by the Central Processing Unit. The complete CPU is allocated to the Process and there would be no change of CPU allocation until the process is complete.

In Non-Pre Emptive-Approach the process once stars its execution, then the CPU is not allotted to the same process until the completion of process. There is a shift of Processes by the Central Processing Unit. The complete CPU is allocated to the Process when only certain required conditions are achieved and there will be change of CPU allocation if there is break or False occurrence in the required conditions.

## Types of CPU Scheduling Algorithms

- First Come First Serve
- Shortest Job First
- Priority Scheduling
- Round Robin Scheduling

## First Come First Serve Scheduling Algorithm

This is the first type of CPU Scheduling Algorithms. Here, in this CPU Scheduling Algorithm we are going to learn how CPU is going to allot resources to the certain process.

Here, in the First Come First Serve CPU Scheduling Algorithm, the CPU allots the resources to the process in a certain order. The order is serial way. The CPU is allotted to the process in which it has occurred.

We can also say that First Come First Serve CPU Scheduling Algorithm follows First In First Out in Ready Queue.

First Come First Serve can be called as FCFS in short form.

### **Characteristics of FCFS (First Come First Serve):**

- First Come First Serve can follow or can be executed in Pre emptive Approach or Non-Pre emptive Approach
- The Process which enters the Ready Queue is executed First. So, we say that FCFS follows First in First Out Approach.
- First Come First Come First Serve is only executed when the Arrival Time (AT) is greater than or equal to the Time which is at present.

### **Advantages**

- Very easy to perform by the CPU
- Follows FIFO Queue Approach

### **Disadvantages**

- First Come First Serve is not very efficient.
- First Come First Serve suffers because of Convoy Effect.

### **Examples**

1.	Process ID	Process Name	Arrival Time	Burst Time
2.	-----	-----	-----	-----
3.	P 1	A	0	6
4.	P 2	B	2	2
5.	P 3	C	3	1
6.	P 4	D	4	9
7.	P 5	E	5	8

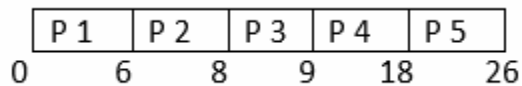
Now, we are going to apply FCFS (First Come First Serve) CPU Scheduling Algorithm for the above problem.

In FCFS, there is an exception that Arrival Times are not removed from the Completion Time.

Here, in First Come First Serve the basic formulas do not work. Only the basic formulas work.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P 1	0	6	6	6	0
P 2	2	2	8	8	6
P 3	3	1	9	9	8
P4	4	9	18	18	9
P 5	5	8	26	26	18

## Gantt Chart



**Average Completion Time** = The Total Sum of Completion Times which is divided by the total number of processes is known as Average Completion Time.

$$\text{Average Completion Time} = (CT_1 + CT_2 + CT_3 + \dots + CT_n) / n$$

### Average Completion Time

1. Average **CT** =  $(6 + 8 + 9 + 18 + 26) / 5$
2. Average **CT** =  $67 / 5$
3. Average **CT** =  $13.4$

**Average Turn Around Time** = The Total Sum of Turn Around Times which is divided by the total number of processes is known as Average Turn Around Time.

$$\text{Average Turn Around Time} = (TAT_1 + TAT_2 + TAT_3 + \dots + TAT_n) / n$$

### Average Turn Around Time

1. Average **TAT** =  $(6 + 8 + 9 + 18 + 26) / 5$

2. Average **TAT** = 13.4

**Average Waiting Time** = The Total Sum of Waiting Times which is divided by the total number of processes is known as Average Waiting Time.

$$\text{Average Waiting Time} = (WT_1 + WT_2 + WT_3 + \dots + WT_n) / n$$

### Average Waiting Time

1. Average **WT** = ( 0 + 6 + 8 + 9 + 18 ) / 5
2. Average **WT** = 41 / 5
3. Average **WT** = 8.2

## Convoy effect

### Examples:

1.	S. No	ProcessID	Process Name	Arrival Time	Burst Time
2.	---	-----	-----	-----	-----
3.	1	P 1	A	1	79
4.	2	P 2	B	0	2
5.	3	P 3	C	1	3
6.	4	P 4	D	0	1
7.	5	P 5	E	2	25

S. No	Process Id	Process Name	Arrival Time ( AT )	Burst Time ( BT )	Completion Time ( CT )	Turn Around Time ( TAT )	Waiting Time ( WT )	Response Time ( RT )
1	P 1	A	1	79	79	78	0	0
2	P 2	B	0	2	81	81	79	79
3	P 3	C	1	3	84	83	81	81
4	P 4	D	0	1	85	85	84	84
5	P 5	E	2	25	110	108	85	85
6	P 6	F	3	3	113	110	110	110

The Average Completion Time is:

Average CT = ( 79 + 81 + 84 + 85 + 110 + 113 ) / 6

Average CT = 92

**The Average Waiting Time is:**

Average WT = ( 0 + 79 + 81 + 84 + 85 + 110 ) / 6

Average WT = 73.1666667

**The Average Turn Around Time is:**

Average TAT = ( 78 + 81 + 83 + 85 + 108 + 110 ) / 6

Average TAT = 90.833334

The biggest problem in this is higher Completion Time, higher Waiting Time, higher Turn Around Time, lower efficiency.

The problem which we found in the above solution can be resolved by using a new CPU Scheduling Techniques named Shortest Job First (SJF).

## Shortest Job First CPU Scheduling Algorithm

This is another type of CPU Scheduling Algorithms. Here, in this CPU Scheduling Algorithm we are going to learn how CPU is going to allot resources to the certain process.

The Shortest Job is heavily dependent on the Burst Times. Every CPU Scheduling Algorithm is basically dependent on the Arrival Times. Here, in this Shortest Job First CPU Scheduling Algorithm, the CPU allots its resources to the process which is in ready queue and the process which has least Burst Time.

If we face a condition where two processes are present in the Ready Queue and their Burst Times are same, we can choose any process for execution. In actual Operating Systems, if we face the same problem then sequential allocation of resources takes place.

Shortest Job First can be called as SJF in short form.

**Characteristics:**

- SJF (Shortest Job First) has the least average waiting time. This is because all the heavy processes are executed at the last. So, because of this reason all the very small, small processes are executed first and prevent starvation of small processes.
- It is used as a measure of time to do each activity.

- If shorter processes continue to be produced, hunger might result. The idea of aging can be used to overcome this issue.
- Shortest Job can be executed in Pre emptive and also non pre emptive way also.

### Advantages

- SJF is used because it has the least average waiting time than the other CPU Scheduling Algorithms
- SJF can be termed or can be called as long term CPU scheduling algorithm.

### Disadvantages

- Starvation is one of the negative traits Shortest Job First CPU Scheduling Algorithm exhibits.
- Often, it becomes difficult to forecast how long the next CPU request will take

### Examples for Shortest Job First

1.	Process ID	Arrival Time	Burst Time
2.	-----	-----	-----
3.	P0	1	3
4.	P1	2	6
5.	P2	1	2
6.	P3	3	7
7.	P4	2	4
8.	P5	5	5

Now, we are going to solve this problem in both pre emptive and non pre emptive way

We will first solve the problem in non pre emptive way.

In Non pre emptive way, the process is executed until it is completed.

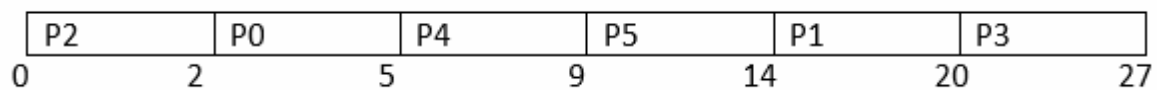
### Non Pre Emptive Shortest Job First CPU Scheduling

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time TAT = CT - AT	Waiting Time WT = CT - BT
P0	1	3	5	4	1



P1	2	6	20	18	12
P2	0	2	2	2	0
P3	3	7	27	24	17
P4	2	4	9	7	4
P5	5	5	14	10	5

#### Gantt Chart:



Now let us find out Average Completion Time, Average Turn Around Time, Average Waiting Time.

#### Average Completion Time:

1. Average Completion Time =  $(5 + 20 + 2 + 27 + 9 + 14) / 6$
2. Average Completion Time =  $77 / 6$
3. Average Completion Time =  $12.833$

#### Average Waiting Time:

1. Average Waiting Time =  $(1 + 12 + 17 + 0 + 5 + 4) / 6$
2. Average Waiting Time =  $37 / 6$
3. Average Waiting Time =  $6.666$

#### Average Turn Around Time:

1. Average Turn Around Time =  $(4 + 18 + 2 + 24 + 7 + 10) / 6$
2. Average Turn Around Time =  $65 / 6$
3. Average Turn Around Time =  $6.166$

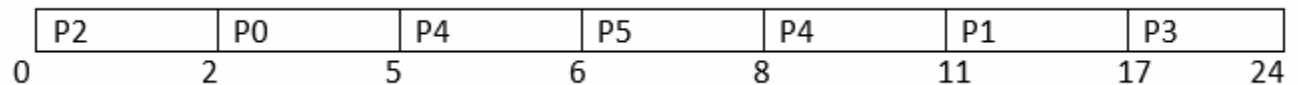
### Pre Emptive Approach

In Pre emptive way, the process is executed when the best possible solution is found.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time TAT = CT -	Waiting Time WT = CT - BT
------------	--------------	------------	-----------------	-----------------------------	---------------------------

AT					
P0	1	3	5	4	1
P1	2	6	17	15	9
P2	0	2	2	2	0
P3	3	7	24	21	14
P4	2	4	11	9	5
P5	6	2	8	2	0

**Gantt chart:**



After P4 P5 is executed just because of the Pre Emptive Condition.

Now let us find out Average Completion Time, Average Turn Around Time, Average Waiting Time.

### Average Completion Time

1. Average Completion Time =  $(5 + 17 + 2 + 24 + 11 + 8) / 6$
2. Average Completion Time =  $67 / 6$
3. Average Completion =  $11.166$

### Average Turn Around Time

1. Average Turn Around Time =  $(4 + 15 + 2 + 21 + 9 + 2) / 6$
2. Average Turn Around Time =  $53 / 6$
3. Average Turn Around Time =  $8.833$

### Average Waiting Time

1. Average Waiting Time =  $(1 + 9 + 0 + 14 + 5 + 0) / 6$
2. Average Waiting Time =  $29 / 6$
3. Average Waiting Time =  $4.833$

## Priority CPU Scheduling

This is another type of CPU Scheduling Algorithms. Here, in this CPU Scheduling Algorithm we are going to learn how CPU is going to allot resources to the certain process.

The Priority CPU Scheduling is different from the remaining CPU Scheduling Algorithms. Here, each and every process has a certain Priority number.

There are two types of Priority Values.

- Highest Number is considered as Highest Priority Value
- Lowest Number is considered as Lowest Priority Value

**Priority for Prevention** The priority of a process determines how the CPU Scheduling Algorithm operates, which is a preemptive strategy. Since the editor has assigned equal importance to each function in this method, the most crucial steps must come first. The most significant CPU planning algorithm relies on the FCFS (First Come First Serve) approach when there is a conflict, that is, when there are many processors with equal value.

### **Characteristics**

- Priority CPU scheduling organizes tasks according to importance.
- When a task with a lower priority is being performed while a task with a higher priority arrives, the task with the lower priority is replaced by the task with the higher priority, and the latter is stopped until the execution is finished.
- A process's priority level rises as the allocated number decreases.

### **Advantages**

- The typical or average waiting time for Priority CPU Scheduling is shorter than First Come First Serve (FCFS).
- It is easier to handle Priority CPU scheduling
- It is less complex

### **Disadvantages**

- The Starvation Problem is one of the Pre emptive Priority CPU Scheduling Algorithm's most prevalent flaws. Because of this issue, a process must wait a longer period of time to be scheduled into the CPU. The hunger problem or the starvation problem is the name given to this issue.

### **Examples:**

Now, let us explain this problem with the help of an example of Priority Scheduling

1.	S. No	Process ID	Arrival Time	Burst Time	Priority
2.	-----	-----	-----	-----	-----
3.	1	P 1	0	5	5
4.	2	P 2	1	6	4
5.	3	P 3	2	2	0
6.	4	P 4	3	1	2
7.	5	P 5	4	7	1
8.	6	P 6	4	6	3

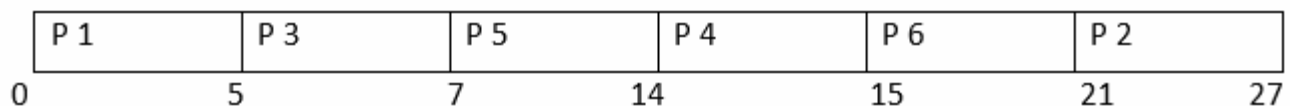
Here, in this problem the priority number with highest number is least prioritized.

This means 5 has the least priority and 0 has the highest priority.

**Solution:**

S. No	Process Id	Arrival Time	Burst Time	Priority	Completion Time Turn Around Time TAT = CT - AT	Waiting Time WT = TAT - BT	
1	PP 1	P0	P5	P5	P5	P5	P0
2	PP 2	P1	P6	P4	P27	P26	P20
3	PP 3	P2	P2	P0	P7	P5	P3
4	PP 4	P3	P1	P2	P15	P12	P11
5	PP 5	P4	P7	P1	P14	P10	P3
6	P 6	P4	P6	P3	P21	P17	P11

**Gantt Chart:**



**Average Completion Time**

1. Average Completion Time =  $(5 + 27 + 7 + 15 + 14 + 21) / 6$
2. Average Completion Time =  $89 / 6$
3. Average Completion Time =  $14.8333$

#### Average Waiting Time

1. Average Waiting Time =  $(0 + 20 + 3 + 11 + 3 + 11) / 6$
2. Average Waiting Time =  $48 / 6$
3. Average Waiting Time =  $7$

#### Average Turn Around Time

1. Average Turn Around Time =  $(5 + 26 + 5 + 11 + 10 + 17) / 6$
2. Average Turn Around Time =  $74 / 6$
3. Average Turn Around Time =  $12.333$

## Round Robin CPU Scheduling

Round Robin is a CPU scheduling mechanism that cycles around assigning each task a specific time slot. It is the First come, First served CPU Scheduling technique with preemptive mode. The Round Robin CPU algorithm frequently emphasizes the Time-Sharing method.

#### Round Robin CPU Scheduling Algorithm characteristics include:

- Because all processes receive a balanced CPU allocation, it is straightforward, simple to use, and starvation-free.
- One of the most used techniques for CPU core scheduling. Because the processes are only allowed access to the CPU for a brief period of time, it is seen as preemptive.

#### The benefits of round robin CPU Scheduling:

- Every process receives an equal amount of CPU time, therefore round robin appears to be equitable.
- To the end of the ready queue is added the newly formed process.

#### Examples:

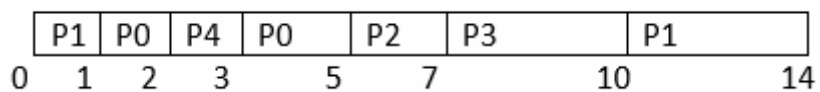
#### Problem

Process ID	Arrival Time	Burst Time
P0	1	3
P1	0	5
P2	3	2
P3	4	3
P4	2	1

**Solution:**

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P0	1	3	5	4	1
P1	0	5	14	14	9
P2	3	2	7	4	2
P3	4	3	10	6	3
P4	2	1	3	1	0

**Gantt Chart:**



**Average Completion Time = 7.8**

**Average Turn Around Time = 4.8**

**Average Waiting Time = 3**

This is all about the CPU Scheduling Algorithms in Operating System