CPU SCHEDULING

First Come First Serve CPU Process Scheduling in Operating Systems

Concept name is First Come First Serve. This is the basic algorithm which every student must learn to understand all the basics of CPU Process Scheduling Algorithms.

First Come First Serve paves the way for understanding of other algorithms. This algorithm may have many disadvantages. But these disadvantages created very new and efficient algorithms. So, it is our responsibility to learn about First Come First Serve CPU Process Scheduling Algorithms.

Important Abbreviations

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

First Come First Serve

First Come First Serve CPU Scheduling Algorithm shortly known as FCFS is the first algorithm of CPU Process Scheduling Algorithm. In First Come First Serve Algorithm what we do is to allow the process to execute in linear manner.

This means that whichever process enters process enters the ready queue first is executed first. This shows that First Come First Serve Algorithm follows First in First Out (FIFO) principle.

The First Come First Serve Algorithm can be executed in Pre-Emptive and Non-Pre Emptive manner. Before, going into examples, let us understand what is Pre-Emptive and Non-Pre-Emptive Approach in CPU Process Scheduling.

Pre-Emptive Approach

In this instance of Pre-Emptive Process Scheduling, the OS allots the resources to a Process for a predetermined period of time. The process transitions from running state to ready state or from waiting state to ready state during resource allocation. This switching happens because the CPU may assign other processes precedence and substitute the currently active process for the higher priority process.

Non-Pre-Emptive Approach

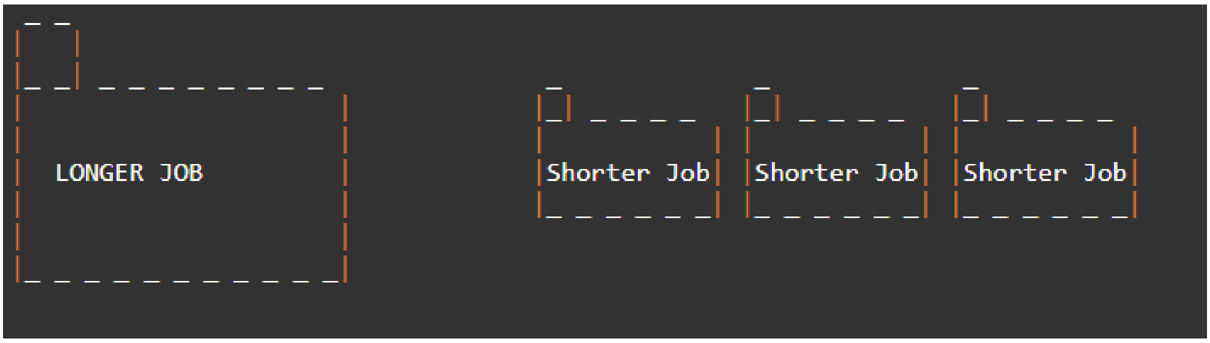
In this case of Non-Pre-Emptive Process Scheduling, the resource cannot be withdrawn from a process before the process has finished running. When a running process finishes and transitions to the waiting state, resources are switched.

Convoy Effect In First Come First Serve (FCFS)

Convoy Effect is a phenomenon which occurs in the Scheduling Algorithm named First Come First Serve (FCFS). The First Come First Serve Scheduling Algorithm occurs in a way of non-pre-emptive way.

The Non pre-emptive way means that if a process or job is started execution, then the operating system must complete its process or job. Until, the process or job is zero the new or next process or job does not start its execution. The definition of Non-Pre-emptive Scheduling in terms of Operating System means that the Central Processing Unit (CPU) will be completely dedicated till the end of the process or job started first and the new process or job is executed only after finishing of the older process or job.

There may be a few cases, which might cause the Central Processing Unit (CPU) to allot a too much time. This is because in the First Come First Serve Scheduling Algorithm Non Pre-emptive Approach, the Processes or the jobs are chosen in serial order. Due, to this shorter jobs or processes behind the larger processes or jobs takes too much time to complete its execution. Due, to this the Waiting Time, Turn Around Time, Completion Time is very high.



```
 _ _
|   |
|_ _| _ _ _ _ _ _ _ _ _ _                  _            _            _
|                         |              |_| _ _ _ _  |_| _ _ _ _  |_| _ _ _ _
|                         |              |           | |           | |           |
|  LONGER JOB             |              |Shorter Job| |Shorter Job| |Shorter Job|
|                         |              |_ _ _ _ _ _| |_ _ _ _ _ _| |_ _ _ _ _ _|
|                         |
|_ _ _ _ _ _ _ _ _ _ _ _ _|
```

So, here as the first process is large or completion time is too high, then this Convoy effect in the First Come First Serve Algorithm is occurred.

Let us assume that Longer Job takes infinite time to complete. Then, the remaining processes have to wait for the same infinite time. Due to this Convoy Effect created by the Longer Job the Starvation of the waiting processes increases very rapidly. This is the biggest disadvantage of FCFS CPU Process Scheduling.

Characteristics of FCFS CPU Process Scheduling

The characteristics of FCFS CPU Process Scheduling are:

1. Implementation is simple.
2. Does not cause any causalities while using
3. It adopts a non-pre-emptive and pre-emptive strategy.
4. It runs each procedure in the order that they are received.
5. Arrival time is used as a selection criterion for procedures.

Advantages of FCFS CPU Process Scheduling

The advantages of FCFS CPU Process Scheduling are:

1. In order to allocate processes, it uses the First In First Out queue.
2. The FCFS CPU Scheduling Process is straight forward and easy to implement.
3. In the FCFS situation pre-emptive scheduling, there is no chance of process starving.
4. As there is no consideration of process priority, it is an equitable algorithm.

Disadvantages of FCFS CPU Process Scheduling

The disadvantages of FCFS CPU Process Scheduling are:

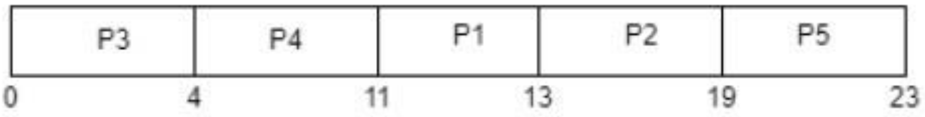
- FCFS CPU Scheduling Algorithm has Long Waiting Time
- FCFS CPU Scheduling favours CPU over Input or Output operations
- In FCFS there is a chance of occurrence of Convoy Effect
- Because FCFS is so straight forward, it often isn't very effective. Extended waiting periods go hand in hand with this. All other orders are left idle if the CPU is busy processing one time-consuming order.

Problems in the First Come First Serve CPU Scheduling Algorithm

Problem 1

Consider the given table below and find Completion time (CT), Turn-around time (TAT), Waiting time (WT), Response time (RT), Average Turn-around time and Average Waiting time.

Process ID	**Arrival time**	**Burst time**
P1	2	2
P2	5	6
P3	0	4
P4	0	7
P5	7	4



Solution

Gantt chart

For this problem CT, TAT, WT, RT is shown in the given table −

Process ID	**Arrival time**	**Burst time**	**CT**	**TAT=CT-AT**	**WT=TAT-BT**	**RT**
P1	2	2	13	13-2= 11	11-2= 9	9
P2	5	6	19	19-5= 14	14-6= 8	8
P3	0	4	4	4-0= 4	4-4= 0	0
P4	0	7	11	11-0= 11	11-7= 4	4
P5	7	4	23	23-7= 16	16-4= 12	12

Average Waiting time = (9+8+0+4+12)/5 = 33/5 = 6.6 time unit (time unit can be considered as milliseconds)

Average Turn-around time = (11+14+4+11+16)/5 = 56/5 = 11.2 time unit (time unit can be considered as milliseconds)

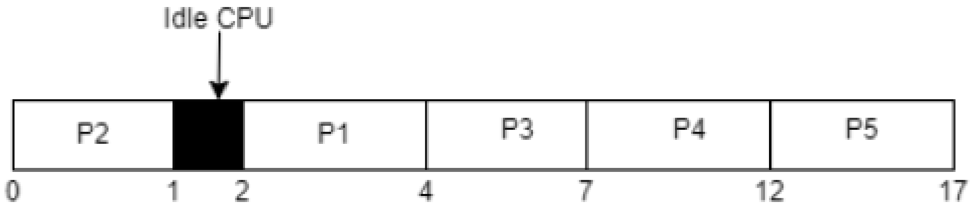
Problem 2

Consider the given table below and find Completion time (CT), Turn-around time (TAT), Waiting time (WT), Response time (RT), Average Turn-around time and Average Waiting time.

Process ID	**Arrival time**	**Burst time**
P1	2	2
P2	0	1
P3	2	3
P4	3	5
P5	4	5

Solution

Gantt chart −



For this problem CT, TAT, WT, RT is shown in the given table −

Process ID	Arrival time	Burst time	CT	TAT=CT-AT	WT=TAT-BT	RT
P1	2	2	4	4-2= 2	2-2= 0	0
P2	0	1	1	1-0= 1	1-1= 0	0
P3	2	3	7	7-2= 5	5-3= 2	2
P4	3	5	12	12-3= 9	9-5= 4	4
P5	4	5	17	17-4= 13	13-5= 8	8

Average Waiting time = (0+0+2+4+8)/5 = 14/5 = 2.8 time unit (time unit can be considered as milliseconds)

Average Turn-around time = (2+1+5+9+13)/5 = 30/5 = 6 time unit (time unit can be considered as milliseconds)

*In idle (not-active) CPU period, no process is scheduled to be terminated so in this time it remains void for a little time.

Problem-03:

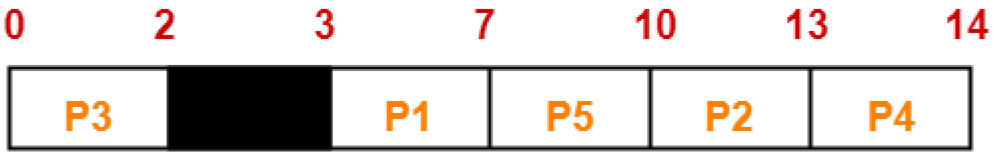
Consider the set of 5 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	3	4
P2	5	3
P3	0	2
P4	5	1
P5	4	3

If the CPU scheduling policy is FCFS, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-



Gantt Chart

Here, black box represents the idle time of CPU.

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Also read- Various Times of Process

Process Id	Exit time	Turn Around time	Waiting time
P1	7	7 – 3 = 4	4 – 4 = 0
P2	13	13 – 5 = 8	8 – 3 = 5
P3	2	2 – 0 = 2	2 – 2 = 0
P4	14	14 – 5 = 9	9 – 1 = 8
P5	10	10 – 4 = 6	6 – 3 = 3

Now,

- Average Turn Around time = (4 + 8 + 2 + 9 + 6) / 5 = 29 / 5 = 5.8 unit
- Average waiting time = (0 + 5 + 0 + 8 + 3) / 5 = 16 / 5 = 3.2 unit