

First come First Come First Serve Scheduling (FCFS)

Ex: 1

Process	Arrival Time	Processing Time
P_1	0	3
P_2	2	3
P_3	3	1
P_4	5	4
P_5	8	2

If the processes arrive as per the arrival time, the Gantt chart will be :

www.allbca.com

	P_1	P_2	P_3	P_4	P_5
0	3	6	7	11	13

Time	Process Complete	Turn Around Time	Processing Time
0	-	-	-
3	P_1	$3 - 0 = 3$	$3 - 3 = 0$
6	P_2	$6 - 2 = 4$	$4 - 3 = 1$
7	P_3	$7 - 1 = 6$	$6 - 1 = 5$
11	P_4	$11 - 4 = 7$	$7 - 4 = 3$
13	P_5	$13 - 2 = 11$	$11 - 2 = 9$

by Mayank Rana

$$\text{Average turn around time} = \frac{(3 + 4 + 6 + 7 + 11)}{5} = 6.2$$

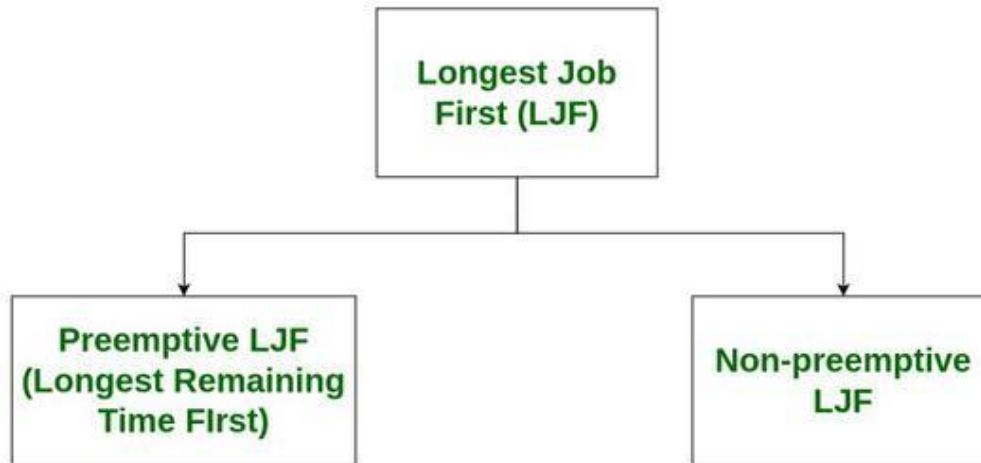
$$\text{Average waiting time} = \frac{(0 + 1 + 5 + 3 + 9)}{5} = 3.6$$

$$\text{Throughput} = \frac{5}{13} = 0.38$$

Ex:2

Process	Arrival Time	Burst Time
p1	1	3
p2	2	3
p3	2	1
p4	4	4
p5	5	2

Types of LJF Scheduling Algorithms



Characteristics of Longest Job First (Non-Preemptive)

- Among all the processes waiting in a waiting queue, the CPU is always assigned to the process having the largest burst time.
- If two processes have the same burst time then the tie is broken using FCFS i.e. the process that arrived first is processed first.
- LJF CPU Scheduling can be of both preemptive and non-preemptive types.

Advantages of Longest Job First(LJF)

- No other process can execute until the longest job or process executes completely.
- All the jobs or processes finish at the same time approximately.

Disadvantages of Longest Job First CPU Scheduling Algorithm

- This algorithm gives a very high average waiting time and average turn-around time for a given set of processes.
- This may lead to a convoy effect.
- It may happen that a short process may never get executed and the system keeps on executing the longer processes.
- It reduces the processing speed and thus reduces the efficiency and utilization of the system.

Longest Job First CPU Scheduling Algorithm

- **Step-1:** First, sort the processes in increasing order of their Arrival Time.
- **Step 2:** Choose the process having the highest Burst Time among all the processes that have arrived till that time.
- **Step 3:** Then process it for its burst time. Check if any other process arrives until this process completes execution.
- **Step 4:** Repeat the above three steps until all the processes are executed.

Ex:1

process	Arrival Time	Burst Time
p1	1	2
p2	2	4
p3	3	6
p4	4	8

P.No.	Arrival Time (AT)	Completion Time (CT)	Burst Time (BT)	Turn Around Time (TAT)	Waiting Time (WT)
P1	1	3	2	2	0
P2	2	21	4	19	15
P3	3	9	6	6	0
P4	4	17	8	13	5

Ex2:

First, let's sort the processes based on their arrival time?

Process	Arrival Time	Burst Time
P1	1	2
P2	2	5
P3	3	3
P4	4	8

Longest Remaining Time First (LRTF) or Preemptive Longest Job First CPU Scheduling Algorithm

Longest Remaining Time First (LRTF) is a preemptive version of Longest Job First (LJF) scheduling algorithm. In this scheduling algorithm, we find the process with the maximum remaining time and then process it, i.e. check for the maximum remaining time after some interval of time(say 1 unit each) to check if another process having more Burst Time arrived up to that time.

Characteristics of Longest Remaining Time First (LRTF)

- Among all the processes waiting in a waiting queue, CPU is always assigned to the process having largest burst time.
- If two processes have the same burst time then the tie is broken using FCFS i.e. the process that arrived first is processed first.
- LJF CPU Scheduling can be of both preemptive and non-preemptive type.

Advantages of Longest Remaining Time First (LRTF)

- No other process can execute until the longest job or process executes completely.
- All the jobs or processes finishes at the same time approximately.

Disadvantages of Longest Remaining Time First (LRTF)

- This algorithm gives very high average waiting time and average turn-around time for a given set of processes.
- This may lead to convoy effect.
- It may happen that a short process may never get executed and the system keeps on executing the longer processes.
- It reduces the processing speed and thus reduces the efficiency and utilization of the system.

Longest Remaining Time First (LRTF) CPU Scheduling Algorithm

Step-1: First, sort the processes in increasing order of their Arrival Time.

Step-2: Choose the process having least arrival time but with most Burst Time.

Step-3: Then process it for 1 unit. Check if any other process arrives up to that time of execution or not.

Step-4: Repeat the above both steps until execute all the processes.

Example1 of Longest Remaining Time First (LRTF)

process	Arrival Time	Burst Time
p1	0	3
p2	1	5
p3	3	7
p4	4	10

Example 2:

Processes	Arrival Time	Burst Time
P1	0ms	2ms
P2	0ms	3ms
P3	2ms	2ms
P4	3ms	5ms
P5	4ms	4ms

Example of Round Robin Scheduling Algorithm

Consider the following 6 processes: **P1, P2, P3, P4, P5, and P6** with their arrival time and burst time as given below:

What are the average waiting and turnaround times for the round-robin scheduling algorithm (RR) with a time quantum of **4 units**?

Process ID	Arrival Time	Burst Time
P1	0	5
P2	1	6
P3	2	3
P4	3	1
P5	4	5
P6	6	6

Advantages of Round Robin Scheduling

- Round Robin scheduling is the most a fair scheduling algorithm whereby all processes are given equal time quantum for execution.
- Starvation of any process caused by indefinite waiting in ready queue is totally eliminated in RR scheduling.
- It does not require any complicated method to calculate the CPU burst time of each process prior to scheduling.
- It is pretty simple to implement and so finds application in a wide range of situations.
- Convoy effect does not occur in RR scheduling as in First Come First Serve CPU (FCFS) scheduling.

Disadvantages of Round Robin Scheduling

- The performance of Round Robin scheduling is highly dependent upon the chosen time quantum. This requires prudent analysis before implementation, failing which required results are not received.
- If the chosen time quantum is very large, most of the processes will complete within the burst time. In effect, RR scheduling will act as FCFS scheduling. Thus, all the limitations of FCFS will come into the system.
- If the chosen time quantum is too small, the CPU will be very busy in context switching, i.e. swapping in swapping out processes to and from the CPU and memory. This would reduce the throughput of the system since more time will be expended in context switching rather than actual execution of the processes.
- RR scheduling does not give any scope to assign priorities to processes. So, system processes which need high priority gets the same preference as background processes. This may often hamper the overall performance of a system.

Example of Round Robin Scheduling

Let us consider a system that has four processes which have arrived at the same time in the order P1, P2, P3 and P4. The burst time in milliseconds of each process is given by the following table –

Process	CPU Burst Times in ms
P1	8
P2	10
P3	6
P4	4

Let us consider time quantum of 2ms and perform RR scheduling on this. We will draw GANTT chart and find the average turnaround time and average waiting time.

GANTT Chart with time quantum of 2ms

