Shortest Job First Scheduling

Shortest job first (SJF) is a scheduling policy that selects the waiting process with the smallest execution time to execute next. This scheduling method can be preemptive or non-preemptive. There are basically two types of SJF methods:

- Non-Preemptive SJF
- Preemptive SJF

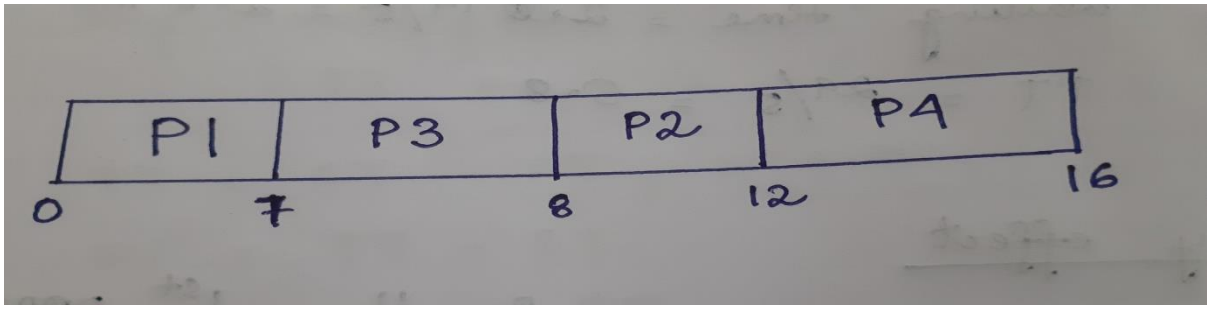
Non-Preemptive SJF

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

Consider the following four processes each having its own unique burst time and arrival time.

Process id	Arrival Time	Burst Time
P1	0	7
P2	2	4
P3	4	1
P4	5	4

Gantt chart :



Process	AT	BT	CT	TT	WT	RT
P1	0	7	7	7	0	0
P2	2	4	12	10	6	6
P3	4	1	8	4	3	3
P4	5	4	16	11	7	7

Here,
AT – Arrival time
BT – Burst time
CT – Completion time
TT – Turnaround time
WT- Waiting time
RT – Response time

Average waiting time= (0+6+3+7)/4 = 16/4 = 4 milliseconds

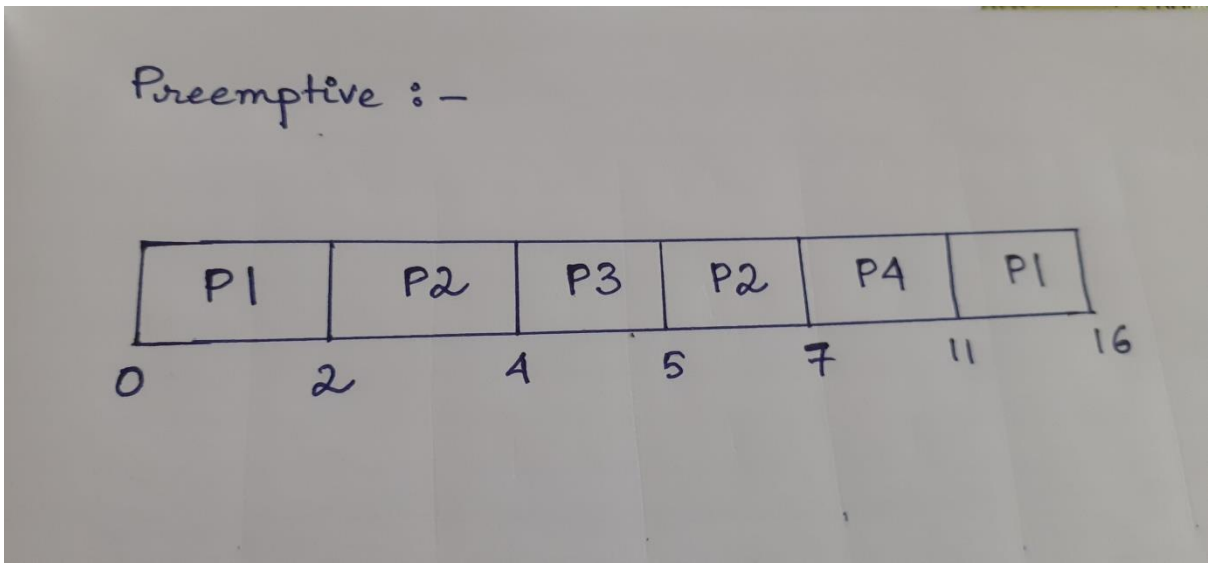
Average turnaround time= (7+10+4+11)/4 = 32/4 = 8 milliseconds

Average response time = (6+3+7)/4 = 4 milliseconds

Preemptive SJF (Shortest Remaining Time First)

In preemptive shortest job first scheduling, jobs are put into ready queue as they arrive, but as a process with short burst time arrives, the existing process is preempted or removed from execution, and the shorter job is executed first. It is also called Shortest Remaining Time First (SRTF) algorithm.

Gantt chart :



Process	AT	BT	CT	TT	WT	RT
P1	0	7	16	16	9	0
P2	2	4	7	5	1	0
P3	4	1	5	1	0	0
P4	5	4	11	6	2	2

Average waiting time= (9+1+0+2)/4 = 12/4 = 3 milliseconds

Average turnaround time= (16+5+1+6)/4 = 28/4 = 7 milliseconds

Average response time = 2/4 = 0.5 milliseconds

Note:

If two processes have same burst time then the tie is broken using FCFS, i.e., the process that arrived first is processed first.

The major advantage of this algorithm is that it gives the minimum waiting time for a given set of processes and thus reduces the average waiting time. But however SJF may cause starvation, if shorter processes keep coming.
