2. Shortest Job First Scheduling (SJF)

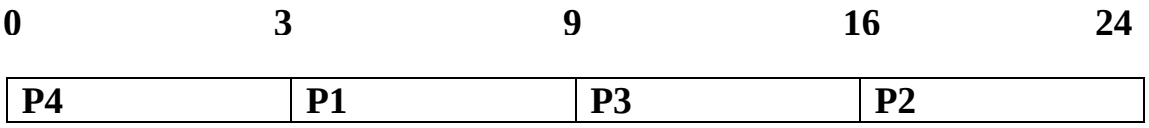
This algorithm associate with each process the length of the latter's next CPU burst. When the CPU is available, it is assigned to the process has the smallest next CPU burst. If two processes have the same length , FCFS scheduling is used to break this tie.

As an example consider the following set of processes with the length of the CPU burst given in millisecond:

Example **3** using **SJF:**

Process	Burst time
P1	6
P2	8
P3	7
P4	3

Gantt chart:



W. T p1 = 3-0 = 3
W. T p2 = 16-0 = 16
W. T p3 = 9-0 = 9
W. T p4 = 0-0 = 0

Average wait time = (3 + 16 + 9 + 0) / 4 = 7 milliseconds
The average completion time = (9+24+16+3)/4 =13 milliseconds
The average waiting time in SJF is the optimal that it gives the minimum average waiting time.

The SJF is either preemptive or non-preemptive.

– ***Non preemptive:*** once CPU given to the process it cannot be preempted until completes its CPU burst.

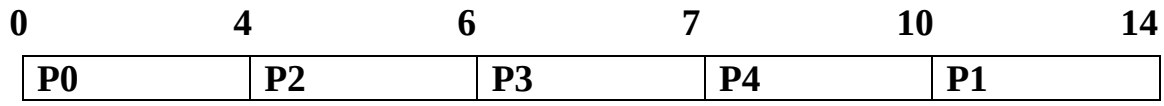
– ***Preemptive:*** if a new process arrives with CPU burst length less than remaining time of current executing process, preempt. This scheme is known as the **Shortest Remaining Time First (SRTF)**.

As an example consider the following set of processes with the length of the CPU burst given in millisecond:

Example **4** using **SJF:**

Process	Burst Time	Arrival Time
P0	4	0
P1	4	1
P2	2	2
P3	1	5
P4	3	7

Gantt chart:



W. T p0 = 0-0 = 0
W. T p1 = 10-1 = 9
W. T p2 = 4-2 = 2
W. T p3 = 6-5 = 1
W. T p4 = 7-7 = 0

Average wait time = (0 + 9 + 2 +1+ 0) / 5 = 2.4 milliseconds

1. **Priority Scheduling Algorithm**

In this algorithm a priority is associated with each process and the CPU is allocated to the process of the highest priority. We use the low numbers to represent high priority.

As an example consider the following set of processes with the length of the CPU burst given in millisecond

Example **5** using **priority**:

Process	Burst time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

Gantt chart:

P2	P5	P1	P3	P4

0 1 6 16 18 19

W. T p1 = 6-0= 6
W. T p2 = 0-0 = 0
W. T p3 = 16-0 = 16
W. T p4 = 18-0 = 18
W. T p5 = 1-0 = 1
Average wait time = (6 + 0 + 16 +18+ 1) / 5 = 8.2 milliseconds

Example **6** using **non-preemptive priority:**

Process	Burst time	Arrival	Priority
P0	4	0	3
P1	2	3	2
P2	5	4	1
P3	6	10	0
P4	8	11	5

Gantt chart:

0 4 9 11 17 25

P0	P2	P1	P3	P4

W. T p0 = 0- 0= 0
W. T p1 = 9-3 = 6
W. T p2 = 4 - 4 = 0
W. T p3 = 11-10 = 1
W. T p4 = 17- 11 = 6

Average wait time = (0 + 6 + 0 +1+ 6) / 5 = 2.6 millisecond

Example **7** using **Preemptive Priority:**

Process	Burst time	Arrival	Priority
P0	5	0	3
P1	8	2	2
P2	3	3	1
P3	6	4	0
P4	2	8	4

Gantt chart:

0 2 3 4 10 12 19 22 24

P0	**P1**	**P2**	**P3**	**P2**	**P1**	**P0**	**P4**

W. T p0 = 19-2-0 = 17
W. T p1 = 12 -2-1 = 9
W. T p2 = 10 – 3- 3 = 4
W. T p3 = 4- 4 = 0
W. T p4 = 22- 8 = 14
Average wait time = (17+9+ 4 + 0 +14) / 5 = 8.8 milliseconds

Priority scheduling can be either Preemptive or non preemptive, when a process arrives the ready queue, its priority is compared with the priority of the currently running process. A preemptive priority will preempt the CPU if the priority of the newly arrived process is higher than the priority of the currently running process. A non preemptive priority scheduling will put the new process with the higher priority than the priority of the currently running process at the head of the ready queue.