Lecture 5

– **Shortest Job First Scheduling Algorithm (SJF)**

In the Shortest Job First scheduling algorithm, jobs (tasks) are scheduled in ascending order of their processing times, i.e., the machine (processor) is allocated to the task with the shortest execution time. SJF can be implemented in two forms: non-preemptive and preemptive.

Non-Preemptive SJF Algorithm

- The scheduler selects the job(task) with the smallest processing time from the ready queue at the moment a machine(processor) becomes available(if the processing times are the same, use arrival order).
- Once starts executing the task, it runs to completion without interruption.

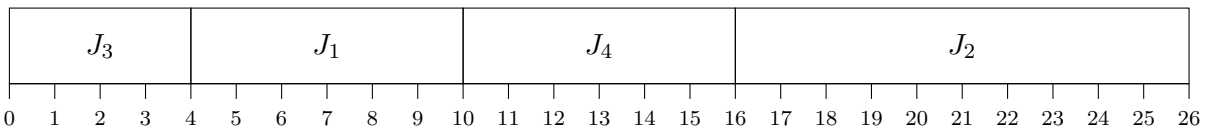
Examples of Non-Preemptive SJF Algorithm

Example 1:

Suppose that we have a set of four jobs that have arrived at the same time in the order J_1, J_2, J_3 and J_4. The processing time in milliseconds of each job is given by the following table

J_i	p_i
J_1	6
J_2	10
J_3	4
J_4	6

Let us draw the Gantt chart and find the average of the waiting time, completion time and Flowtime using non-preemptive SJF algorithm.



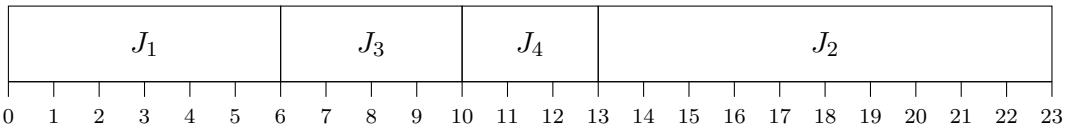
J_i	C_i	W_i	F_i
J_1	10	4	10
J_2	26	16	26
J_3	4	0	4
J_4	16	10	16
Average	14	7.5	14

Example 2:
In the previous example, we had assumed that all the tasks had arrived at the same time, a situation which is practically impossible. Here, we consider the tasks are arrive at different times. Suppose we have a set of four tasks whose arrival times and processing times are as follows

J_i	r_i	p_i
J_1	0	6
J_2	4	10
J_3	4	4
J_4	8	3

Let us draw the Gantt chart and find the average of the waiting time, completion time and Flowtime using non-preemptive SJF algorithm.

While drawing the Gantt chart, we will consider which jobs have arrived in the system when the scheduler is invoked. At time 0ms, only J_1 is there, and so it is assigned to the machine. J_1 completes execution at 6ms, and at that time, J_2 and J_3 have arrived, but not J_4. J_3 is assigned to the machine since it has the shortest processing time among current jobs. J_3 completes execution at time 10ms. By that time, J_4 has arrived, and so SJF algorithm is run on the jobs J_2 and J_4. Hence, we find that the order of execution is J_1, J_3, J_4, J_2 as shown in the following Gantt chart



J_i	C_i	W_i	F_i
J_1	6	0	6
J_2	23	9	19
J_3	10	2	6
J_4	13	2	5
Average	13	3.25	9

Preemptive SJF Algorithm

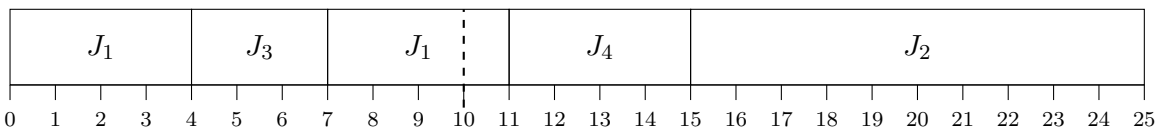
- At any moment, the job (task) with the shortest remaining processing time in the ready queue is executed.
- If a new job (task) arrives with a shorter remaining time than the current job (task), the running job (task) is preempted.

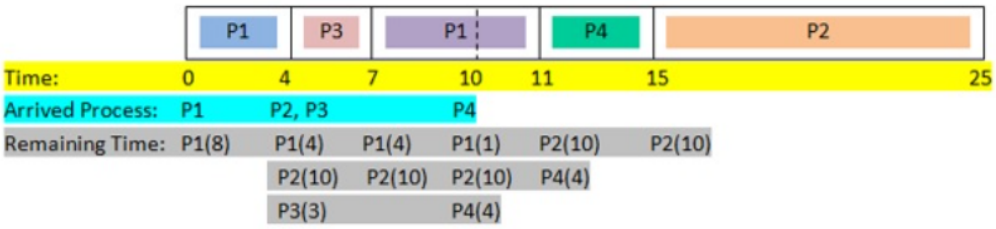
Example of Preemptive SJF algorithm
Let us now perform preemptive SJF scheduling on the following jobs, draw Gantt chart and find the average of the waiting time, completion time and Flowtime.

J_i	r_i	p_i
J_1	0	8
J_2	4	10
J_3	4	3
J_4	10	4

Since this is a preemptive scheduling algorithm, the scheduler is invoked when a job arrives and when it completes execution. The scheduler computes the remaining time to completion for each job in the system and selects the job with the shortest remaining time for execution.

Initially, only J_1 arrives, and so it is assigned to the machine. At time 4ms, J_2 and J_3 arrive. The scheduler computes the remaining time of the jobs J_1, J_2, and J_3 as 4ms, 10ms, and 3ms. Since J_3 has the shortest time, J_1 is preempted by J_3. J_3 completes execution at 7ms, and the scheduler is invoked. Among the jobs in the system, J_1 has the shortest time, and so it executes. At time 10ms, J_4 arrives, and the scheduler again computes the remaining times left for each job. Since the remaining time of J_1 is least, no job switch occurs, and J_1 continues to execute. Similarly, the rest of the jobs complete their execution.





J_i	C_i	W_i	F_i
J_1	11	3	11
J_2	25	11	21
J_3	7	0	3
J_4	15	1	5
Average	14.5	3.75	10