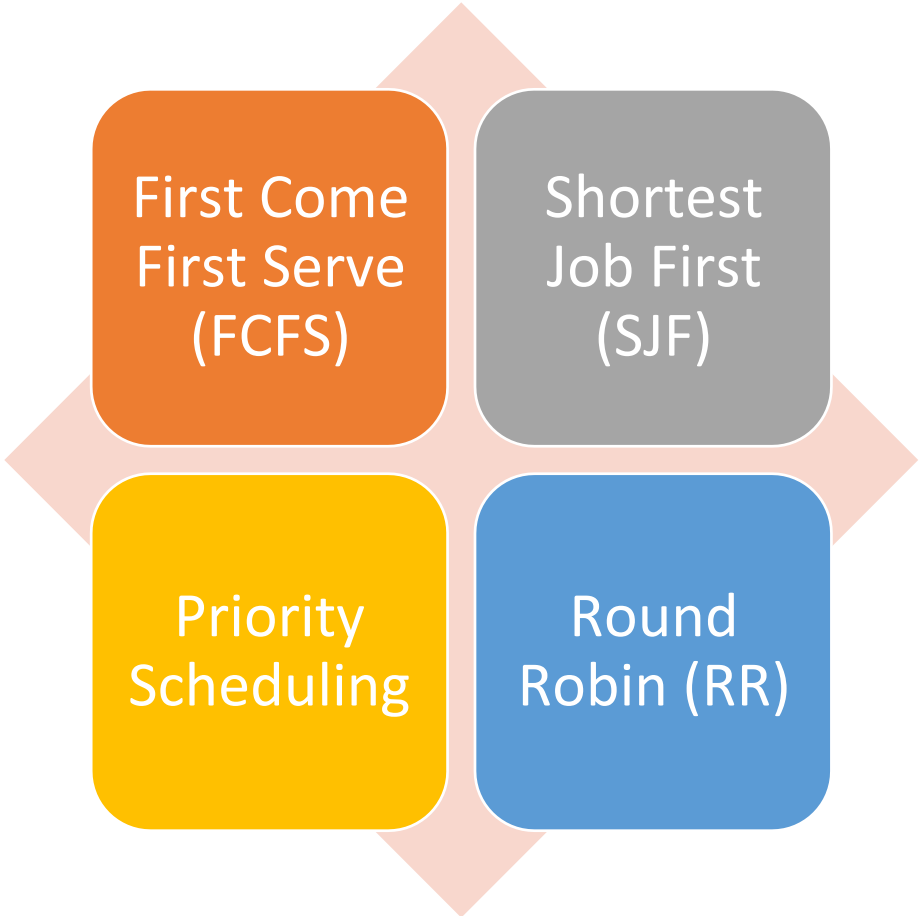
Shortest Job First (SJF)

Nirmala Shinde Baloorkar
Assistant Professor
Department of Computer Engineering

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

CPU Scheduling Algorithms



Shortest-Job-First (SJF) Scheduling

- Associate with each process the length of its CPU burst.
- The CPU is assigned to the process with the smallest/shortest CPU burst time for execution (FCFS can be used to break ties).
- Two schemes:
  - Non preemptive
  - preemptive

Example 1 for Non-Preemptive SJF

Process	Burst time
P1	4
P2	3
P3	5

Turnaround Time = Completion Time – Arrival Time

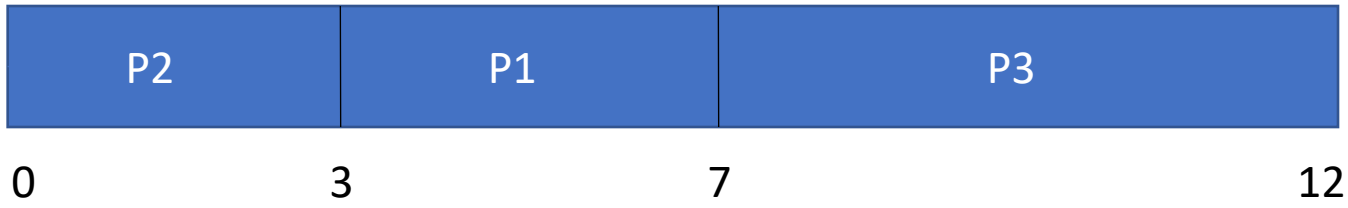
- Gantt Chart

- Turnaround Time p1= 7, p2=3, p3=12

Example 1 for Non-Preemptive SJF

Process	Burst time
P1	4
P2	3
P3	5

- Turnaround Time p1= 7, p2=3, p3=12, Avg. Turnaround Time = 22/3=7.33ms



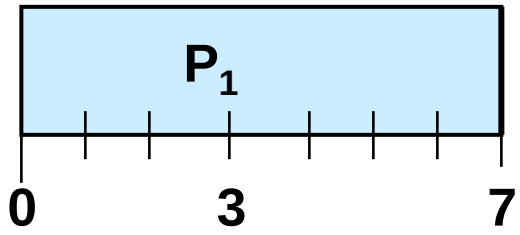
Waiting Time = Turnaround Time – Burst Time

- Waiting Time p1= 3, p2=0, p3=7, Avg. Waiting Time = 10/3=3.33ms

Example 2 for Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- At time 0, P_1 is the only process, so it gets the CPU and runs to completion

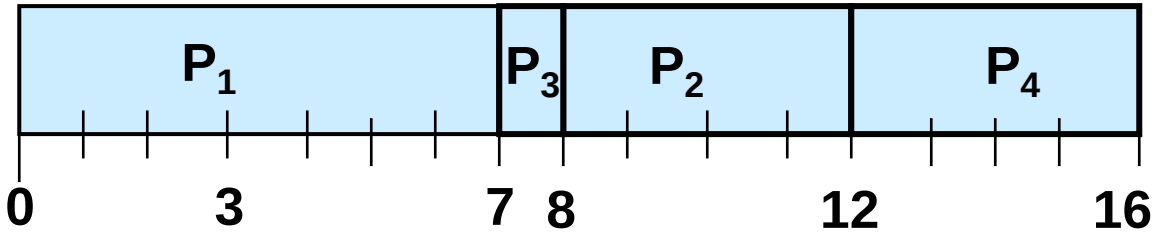


Example for Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

Turnaround Time = Completion Time – Arrival Time

- Once P_1 has completed the queue now holds P_2, P_3 and P_4



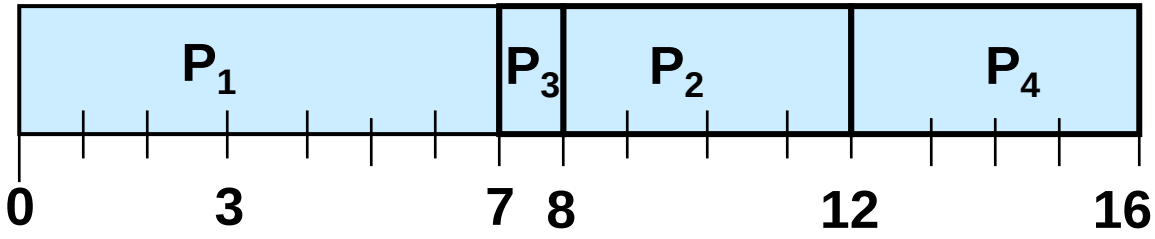
- P_3 gets the CPU first since it is the shortest. P_2 then P_4 get the CPU in turn (based on arrival time)
- Turnaround Time for process p1= 7, p2= 10. p3=4, p4=11
- Average Turnaround time : (7+10+4+11)/4 = 8ms

Example for Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

Waiting Time = Turnaround Time – Burst Time

- Once P_1 has completed the queue now holds P_2, P_3 and P_4



- Turnaround Time for process p1= 7, p2= 10. p3=4, p4=11 Avg. Turnaround Time = 32/4 = 8ms
- Waiting Time for process p1 = 0, p2=6,p3=3,p4=7 Avg. Waiting Time = 16/4 = 4ms

Shortest Job First

Advantages:

- **Optimality**: Minimizes the average waiting time and turnaround time.
- **Efficiency**: Provides efficient CPU utilization in environments where shorter jobs are frequent.

Disadvantages:

- **Starvation**: Longer processes may suffer indefinite postponement if shorter processes keep arriving.
- **Prediction Challenge**: Accurate prediction of burst times is required, which is not always feasible.

Question ?