Chapter 5: CPU Scheduling

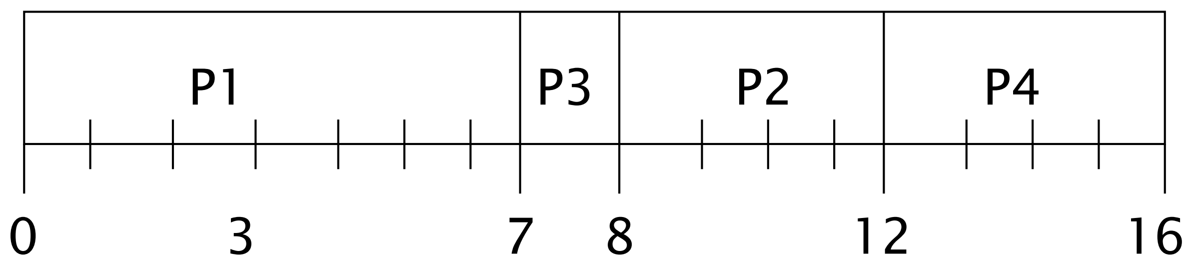
Shortest-Job-First (SJF) Scheduling

- Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time
- Two schemes:
  - **nonpreemptive** – 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 know as the Shortest-Remaining-Time-First (SRTF)
- SJF is optimal – gives minimum average waiting time for a given set of processes

Example of Non-Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

SJF (non-preemptive)

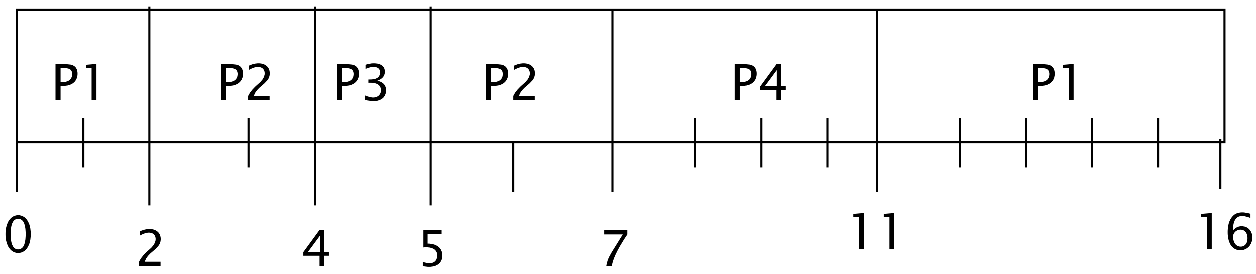


- Waiting times p1=0, p2=6, p3=3, p4=7
- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

Example of Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

SJF (preemptive)



- Waiting times p1=9, p2=1, p3=0, p4=2
- Average waiting time = (9 + 1 + 0 +2)/4 = 3

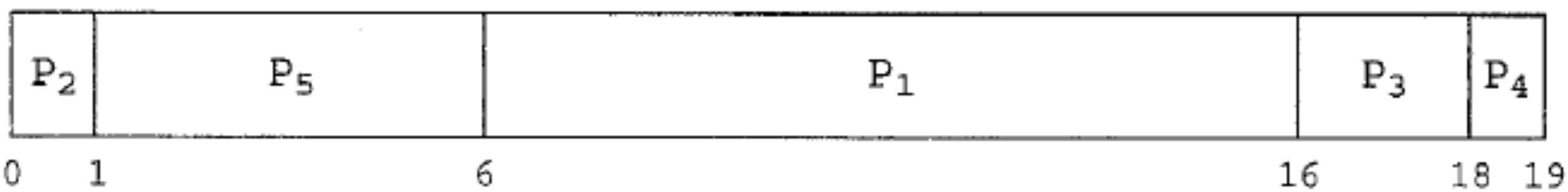
Priority Scheduling

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer ☐ highest priority)
  - Preemptive
  - nonpreemptive
- SJF is a priority scheduling where priority is the predicted next CPU burst time
- **Problem** ☐ Starvation – low priority processes may never execute
- **Solution** ☐ Aging – as time progresses increase the priority of the process

Example of Non-Preemptive Priority scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Waiting times p1=6, p2=0, p3=16, p4=18,p5=1
- Average waiting time = (6 + 0 + 16 +18+1)/5 = 8.2



Round Robin (RR)

- Each process gets a small unit of CPU time (***time quantum***), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.

- If there are n processes in the ready queue and the time quantum is q, then each process gets $1/n$ of the CPU time in chunks of at most q time units at once. No process waits more than $(n-1)q$ time units.

Example of RR with Time Quantum = 20

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

- The Gantt chart is:

P1	P2	P3	P4	P1	P3	P4	P1	P3	P3

0 20 37 57 77 97 117 121 134 154 162

- Waiting times p1=81,p2=20,p3=94,p4=97
- Average waiting time = 73

Round Robin (RR)

- Performance of the RR algorithm depends heavily on the size of the time quantum.
- If the time **quantum is extremely large**, the RR policy is the same as the FCFS policy
- If the **time quantum is extremely small** (say, 1 millisecond), the RR approach is called processor sharing and creates the appearance that each of n processes has its own processor running at 1/n the speed of the real processor. (context switching overhead)

Time Quantum and Context Switch Time