Time Sharing & Network Monitoring System

Time-sharing operating systems

CPU Scheduling

Scheduling Criteria Terminology

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)
- **Burst Time** – actual time that is required to complete execution of particular task or process.

Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P1	24
P2	3
P3	3

Suppose that the processes arrive in the order: P1 , P2 , P3

The Gantt Chart for the schedule is:

First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P1	24
P2	3
P3	3

Suppose that the processes arrive in the order: P1 , P2 , P3
The Gantt Chart for the schedule is:

Waiting time for P1 = 0; P2 = 24; P3 = 27
Average waiting time: (0 + 24 + 27)/3 = 17

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

SJF is optimal – gives minimum average waiting time for a given set of processes

- The difficulty is knowing the length of the next CPU request
- Could ask the user

FCFS Scheduling (Cont.)

Suppose that the processes arrive in the order:

$P2$, $P3$, $P1$

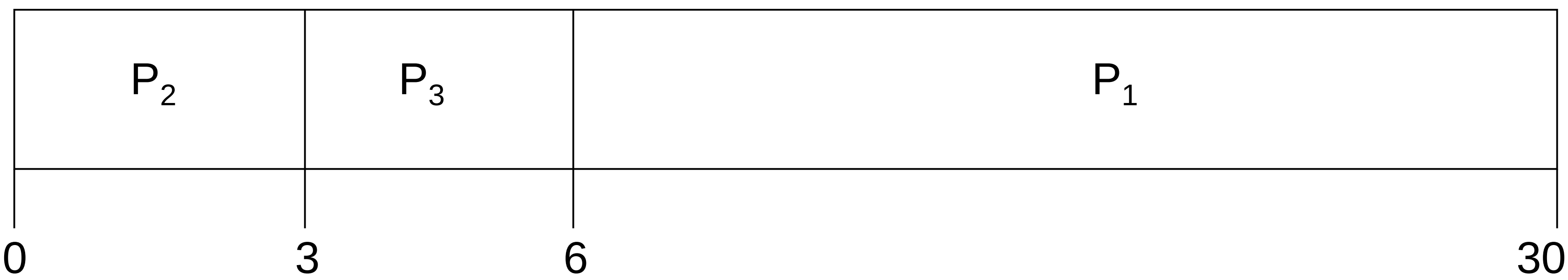
The Gantt chart for the schedule is:

FCFS Scheduling (Cont.)

Suppose that the processes arrive in the order:

$P2$, $P3$, $P1$

The Gantt chart for the schedule is:



Waiting time for $P1 = 6$; $P2 = 0$; $P3 = 3$

Average waiting time: $(6 + 0 + 3)/3 = 3$

Much better than previous case

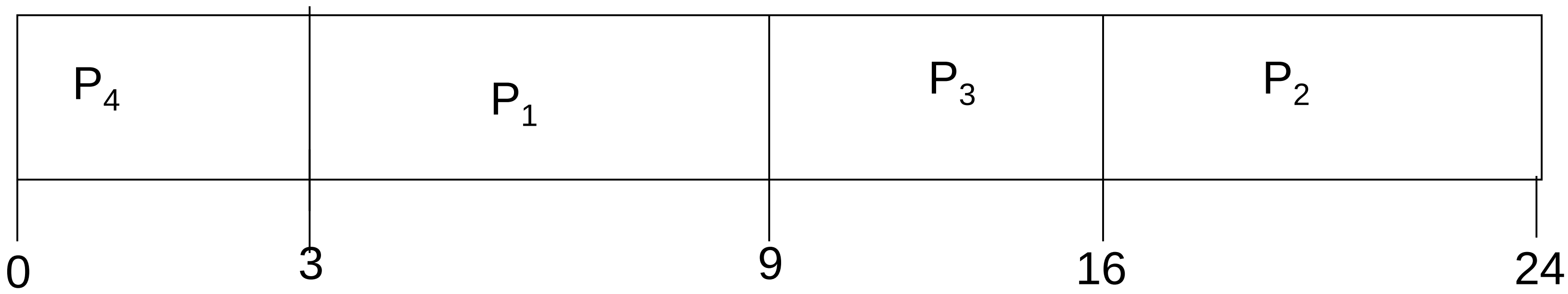
Convoy effect - short process behind long process

Consider one CPU-bound and many I/O-bound processes

Example of SJF

Process	Burst Time
P1	6
P2	8
P3	7
P4	3

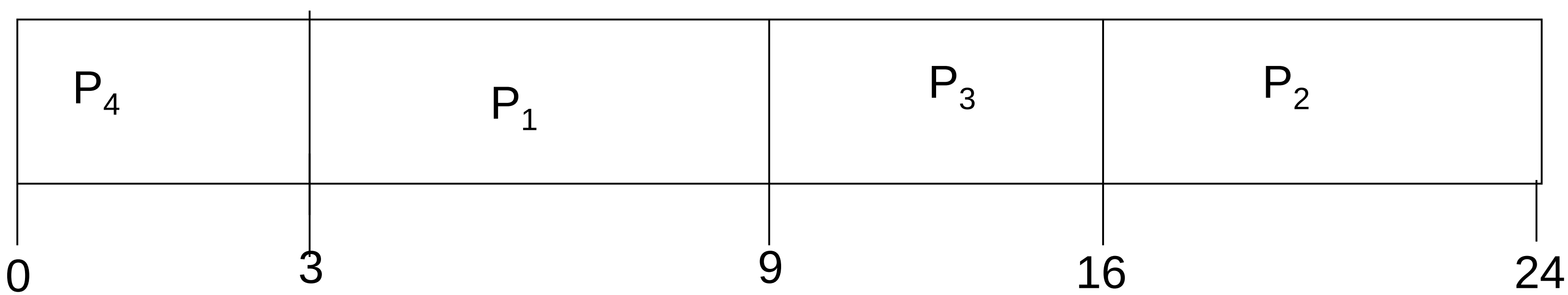
SJF scheduling chart



Example of SJF

Process	Burst Time
P1	6
P2	8
P3	7
P4	3

SJF scheduling chart



Average waiting time = (3 + 16 + 9 + 0) / 4 = 7

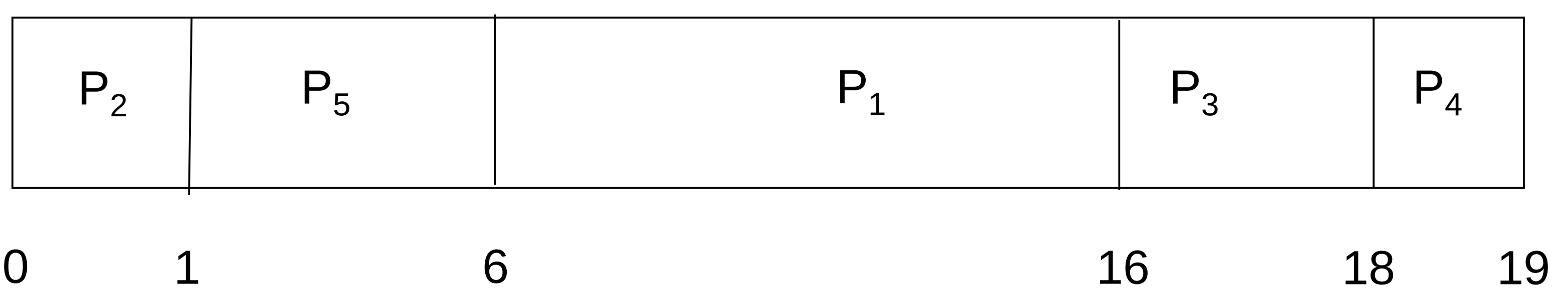
Priority Scheduling

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer = highest priority)
- Problem : Starvation – low priority processes may never execute
- Solution : Aging – as time progresses increase the priority of the process

Example of 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

- Priority scheduling Gantt Chart



- Average waiting time = 8.2

Round Robin (RR)

- Each process gets a small unit of CPU time (time quantum q), usually 10-100 milliseconds. After this time has elapsed, the process is pre-empted and added to the end of the ready queue. The CPU is allocated to the process with the highest priority (smallest integer = highest priority)

- 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.

- Timer interrupts every quantum to schedule next process
- Performance
  - ✓ q large $\Rightarrow$ FIFO
  - ✓ q small $\Rightarrow$ q must be large with respect to context switch, otherwise overhead is too high

Example of RR with Time Quantum = 4

Process	Burst Time
P1	24
P2	3
P3	3

- The Gantt chart is:

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

0 4 7 10 14 18 22 26 30

- Typically, higher average turnaround than SJF, but better response
- q should be large compared to context switch time
- q usually 10ms to 100ms, context switch < 10 usec

Network Monitoring System