Justin Ciocoi

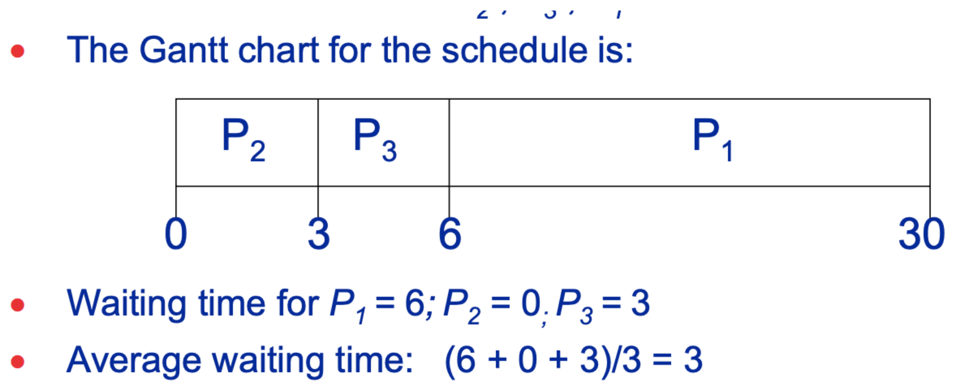
Oct. 12, 2023

CSCI 375 Class Notes

CPU Scheduling

First-Come First-Served Scheduling

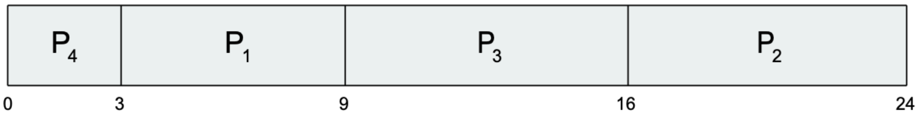
- Fairly self explanatory
- The *convoy effect* refers to the placement of short process after long ones, leading to a higher average waiting time
- **CPU-Bound Process**
  - Process that is bottlenecked by the usage or speed of the CPU - the process "waits" on the CPU
- **I/O-Bound Process**
  - Process that is bottlenecked by the I/O of the system - the process "waits" on the I/O



Shortest Job First Scheduling

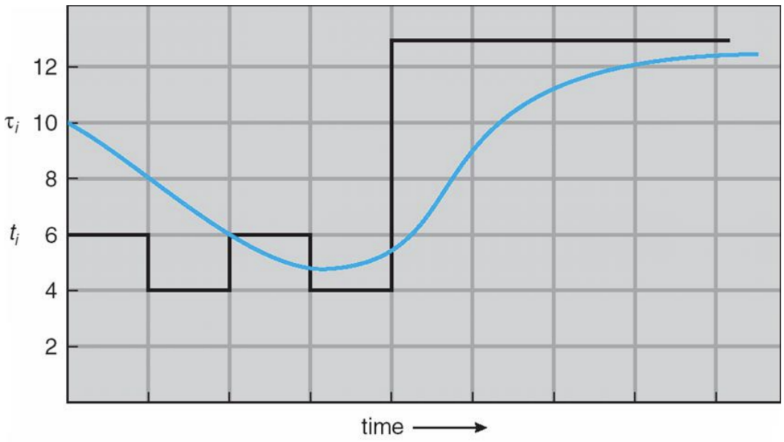
- This guarantees *minimum average waiting time*

- This algorithm is unrealistic to implement in real systems due to the complication of the system not knowing how long each process will take to start



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

- As far as determining the length, in time, of the next CPU length, the best a system can do is estimate it based on previous bursts



CPU burst (t_i)		6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

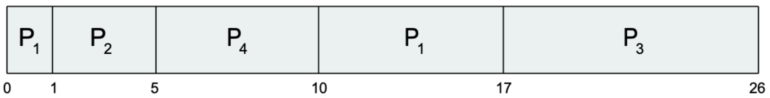
- How is this estimation done?
  - Let t_n be the length of the nth CPU burst
  - τ_{n+1} is the predicted value of the next CPU burst
  - $\alpha, 0 \leq \alpha \leq 1$
  - Define:
    - $\tau_{n+1} = \alpha t_n + (1 - a)\tau_n$
    - commonly, $\alpha = \frac{1}{2}$
    - We can recursively define τ and expand the formula to:
    - $\tau_{n+1} = \alpha t_n + (1 - \alpha)\alpha t_{n-1} + ... + (1 - \alpha)^j \alpha t_{n-j} + ... + (1 - \alpha)^{n+1}\tau_0$

- Since both α and $(1-\alpha)$ are less than or equal to 1, each successive term is less and less weighted than its predecessor
- If α is set to 0, *recent history* does not count
- If α is set to 1, only the single previous CPU burst count

- *Shortest Remaining Time First Scheduling* employs preemptive scheduling to allow for the concepts of varying arrival times and preemption

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



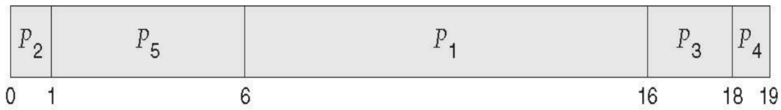
- Average waiting time = [(10-1)+(1-1)+(17-2)+5-3)]/4 = 26/4 = 6.5 msec

Priority Scheduling

- Under priority scheduling, a priority number, which is an integer, is associated with each process
- The CPU is allocated the process with highest priority
  - Can be either *preemptive or non-preemptive*
- Shortest Job First scheduling is a priority scheduling algorithm where the priority is the predicted next CPU burst time
- *Problem: Starvation*
  - Low priority processes might never get the chance to execute
- *Solution: Aging*
  - As time progresses, we can increase the priority of a process such that older processes are more heavily prioritized

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 msec