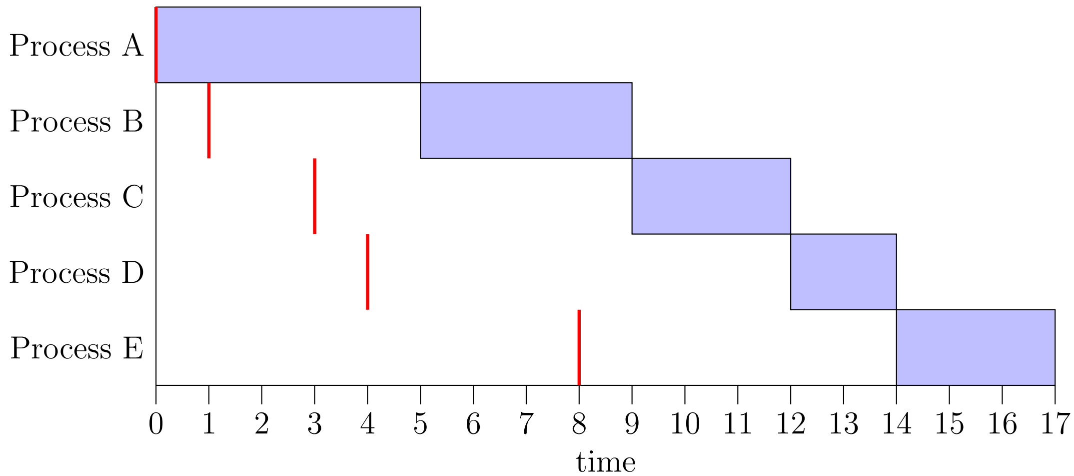
1.) a.) First Come First Served

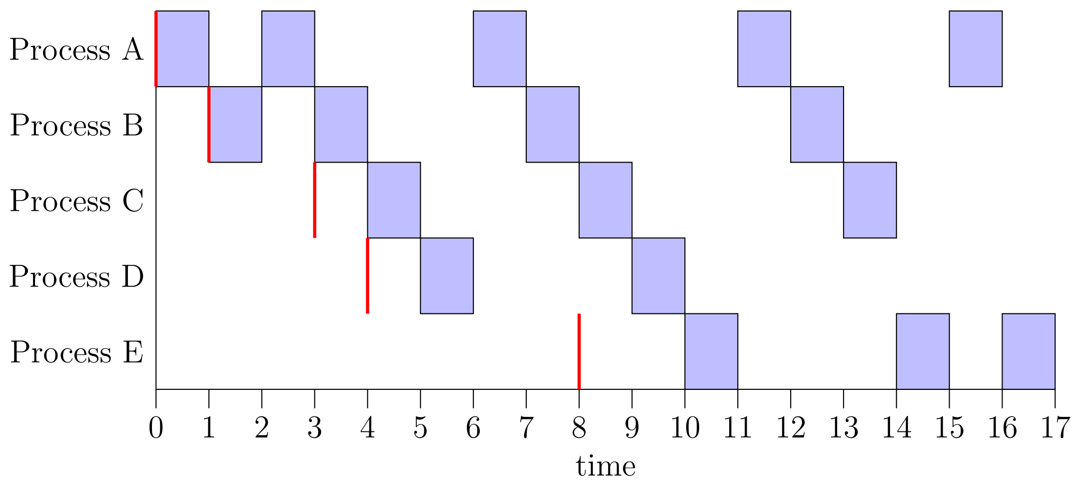


A	B	C	D	E	
5	9	12	14	17	completion time

Average Turnaround: 8.2

Average Normalized Turnaround: 2.8

b.) Round Robin ($q = 1$)

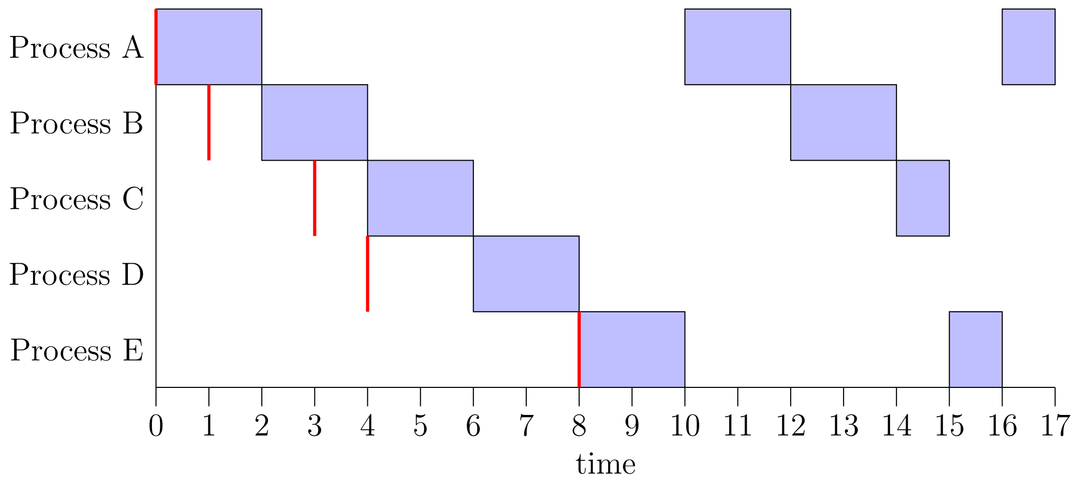


A	B	C	D	E	
16	13	14	10	17	completion time

Average Turnaround: 10.8

Average Normalized Turnaround: 3.173

c.) Round Robin ($q = 2$)

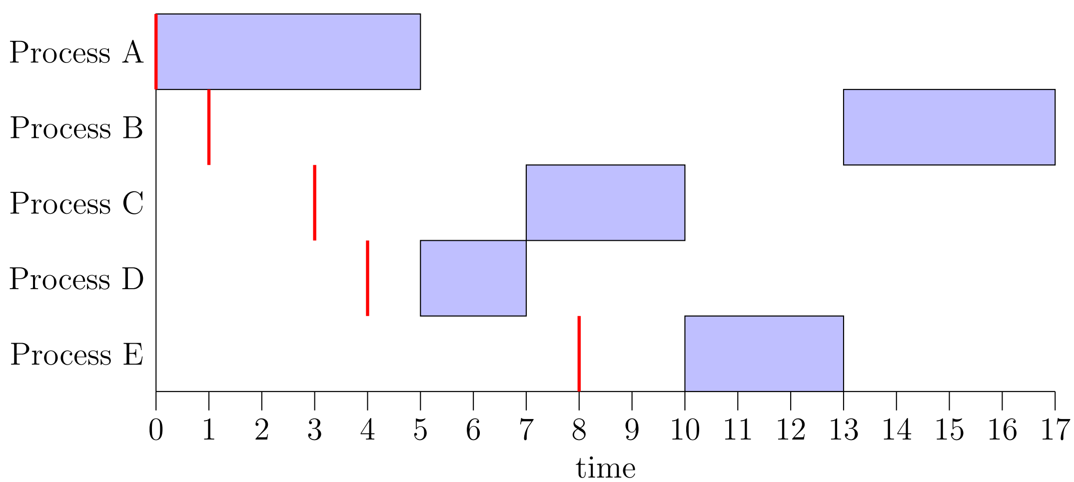


A	B	C	D	E	
17	14	15	8	16	completion time

Average Turnaround: 10.8

Average Normalized Turnaround: 3.063

d.) Shortest Job First

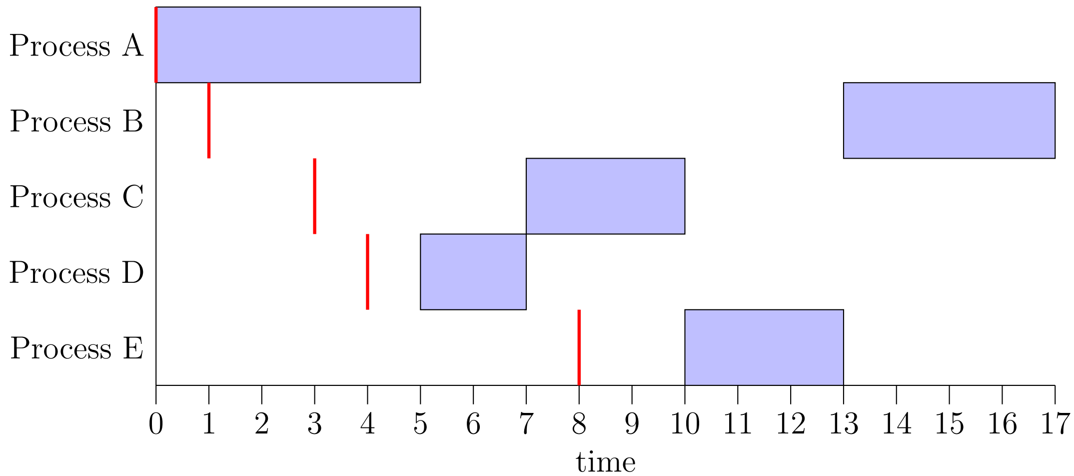


A	B	C	D	E	
5	17	10	7	13	completion time

Average Turnaround: 7.2

Average Normalized Turnaround: 2.1

e.) Shortest Remaining Time First

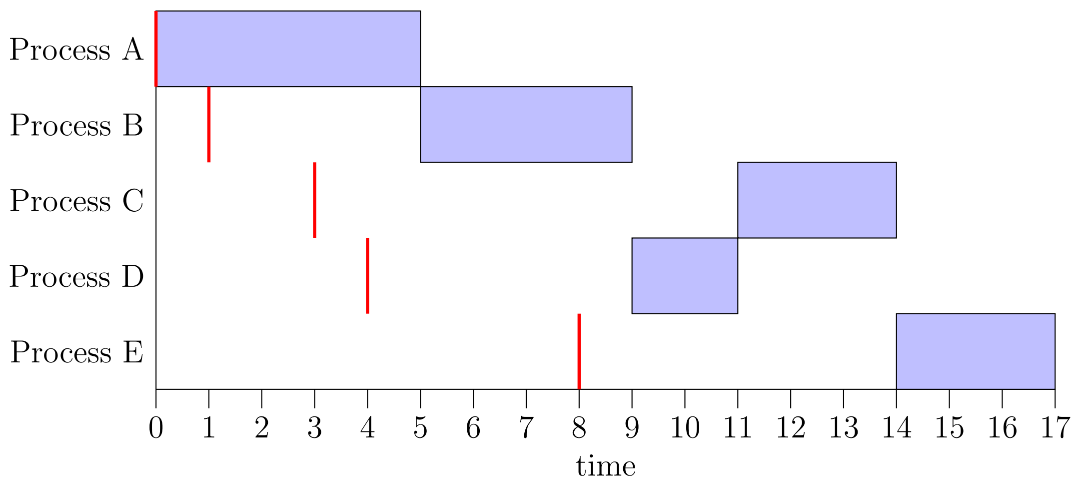


A	B	C	D	E	
5	17	10	7	13	completion time

Average Turnaround: 7.2

Average Normalized Turnaround: 2.1

f.) Highest Response Ratio Next



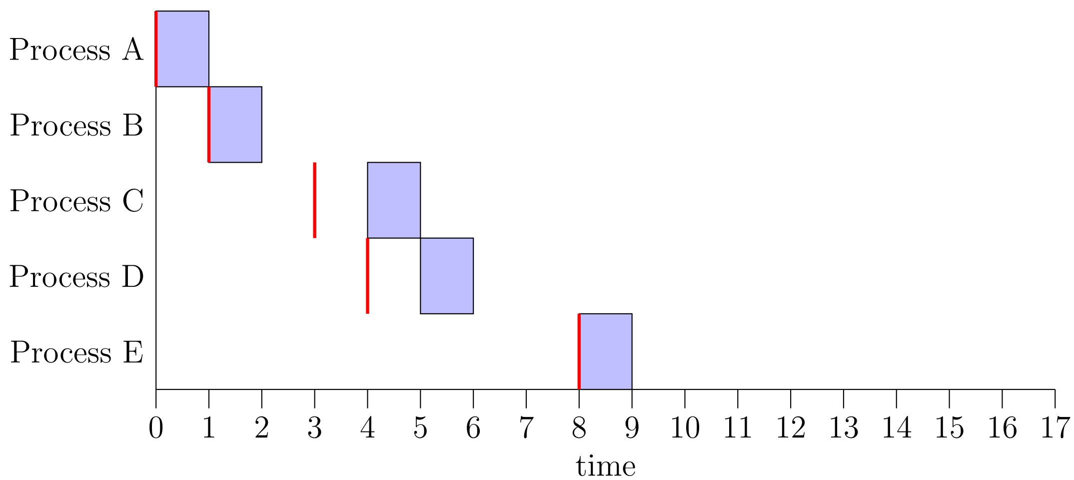
A	B	C	D	E	
5	9	14	11	17	completion time

Average Turnaround: 8

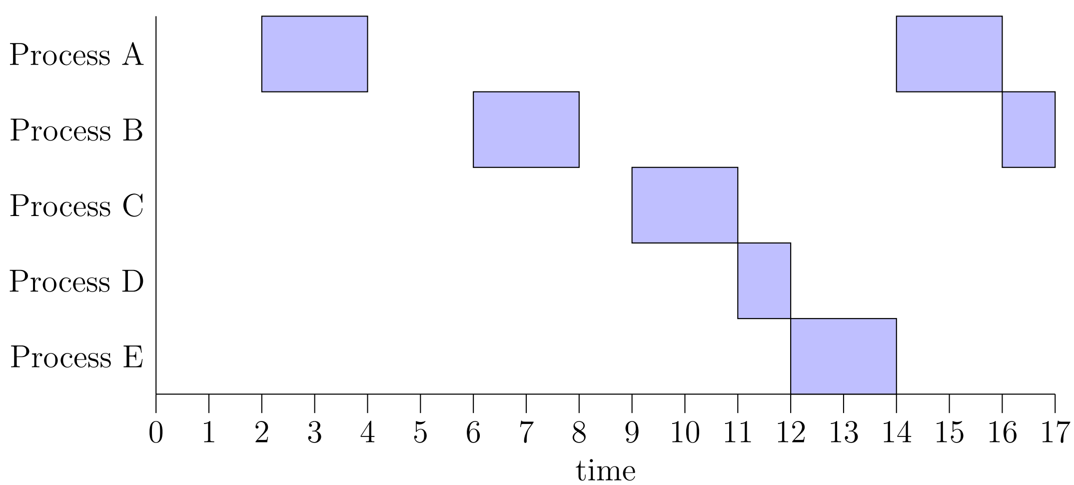
Average Normalized Turnaround: 2.633

g.) Multi-level Feedback With 2 Queues

Queue 1 (First Come First Served, $q = 1$)



Queue 2 (Round Robin, $q = 2$)

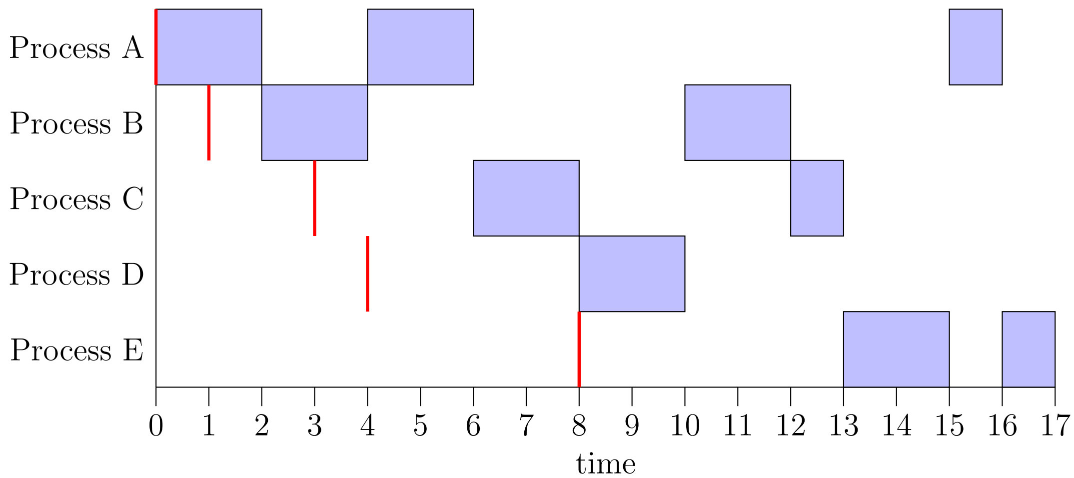


A	B	C	D	E	
16	17	11	12	14	completion time

Average Turnaround: 10.8

Average Normalized Turnaround: 3.173

h.) Preemptive Highest Response Ratio Next ($q = 2$)



A	B	C	D	E	
16	12	13	10	17	completion time

Average Turnaround: 10.4

Average Normalized Turnaround: 3.057

2.) Let $S = \{s_1, s_2, \ldots, s_n\}$ be an arbitrarily ordered set of n process service times. We know that the average total wait time μ_S for S is given by

$$\mu_S = \frac{1}{n}\left(\sum_{k=1}^{n} s_k(n-k)\right).$$

Now suppose we have consecutive $i, j \in \mathbb{N}$ where $1 \le i < j \le n$ and $s_i \ge s_j$, and let S_1 have the same ordering as S but with s_i and s_j swapped, then the new average total wait time μ_{S_1} is given by

$$\mu_{S_1} = \frac{1}{n}\left(\sum_{k=1}^{i-1} s_k(n-k) + s_j(n-i) + s_i(n-j) + \sum_{k=j+1}^{n} s_k(n-k)\right).$$

The difference $\mu_S - \mu_{S_1}$ is given by

$$\begin{aligned}\mu_S - \mu_{S_1} &= \frac{1}{n}\left[(n-i)(s_i - s_j) + (n-j)(s_j - s_i)\right] \\ &= \frac{1}{n}(j-i)(s_i - s_j) = \frac{s_i - s_j}{n}.\end{aligned}$$

Since $s_i \ge s_j$, we know that $s_i - s_j \ge 0$, thus $\mu_S - \mu_{S_1} \ge 0$, and thus $\mu_S \ge \mu_{S_1}$. Repeating this swapping procedure on S, we eventually obtain a set S' where $i < j \implies s_i \le s_j$, which is the same order as scheduling processes by shortest job first. Since we have proven that $\mu_{S'} \le \mu_S$ for arbitrary S, we know that S', and thus shortest job first achieves the smallest possible average wait time for all processes. ∎

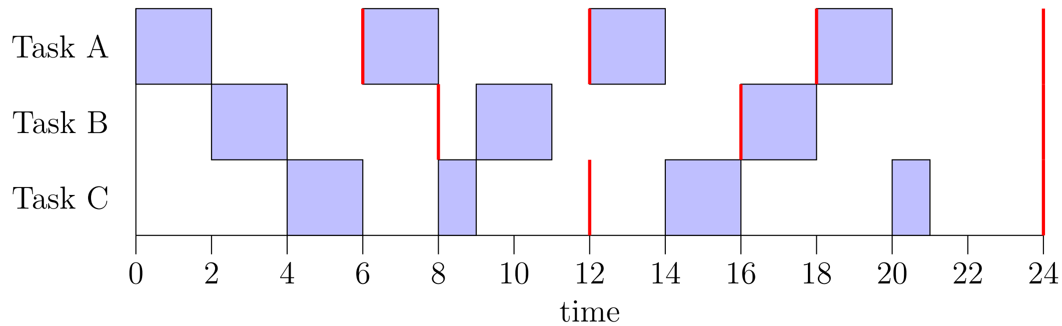
3.) a.) If a task has zero slack, it means that if the task does not start running on the processor immediately, then it will miss its next deadline.

b.) If a task has negative slack, it means that it's impossible for the task to meet its next deadline.

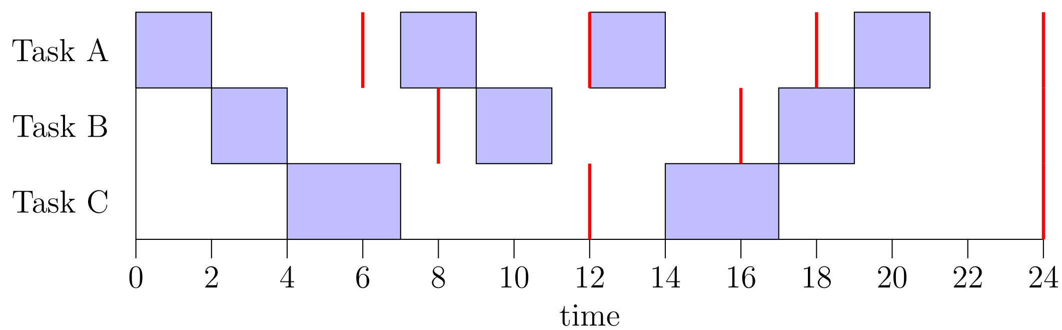
c.) If a task has slack s, it means that the scheduler can delay the task at most s time and still have it meet its next deadline.

d.) (Assuming current process is prioritized when there is a tie)

Least Slack Process Next



Earliest Deadline First



Rate Monotonic Scheduling (priority is inverse to period)