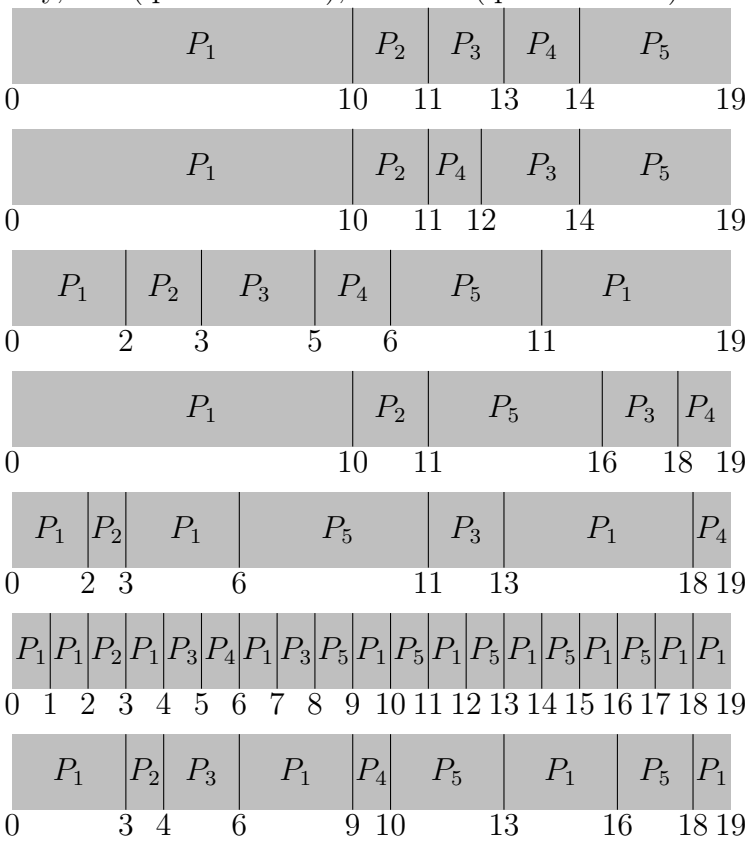
1. (a) The following Gantt charts are in the order of FCFS, nonpreemptive SJF, preemptive SJF, nonpreemptive priority, preemptive priority, RR (quantum = 1), and RR (quantum = 3).



(b)

- FCFS:

$$\text{average waiting time} = \frac{0+8+8+9+8}{5} = 6.6\text{ms}$$

- nonpreemptive SJF:

$$\text{average waiting time} = \frac{0+8+7+9+8}{5} = 6.4\text{ms}$$

- preemptive SJF:

$$\text{average waiting time} = \frac{9+0+0+1+0}{5} = 2\text{ms}$$

- nonpreemptive priority:

$$\text{average waiting time} = \frac{0+8+13+14+5}{5} = 8\text{ms}$$

- preemptive priority:

$$\text{average waiting time} = \frac{8+0+8+14+0}{5} = 6\text{ms}$$

- RR (quantum = 1):

$$\text{average waiting time} = \frac{18}{5} = 3.6\text{ms}$$

- RR (quantum = 3):

$$\text{average waiting time} = \frac{23}{5} = 4.6\text{ms}$$

(c)
- FCFS:

$$\text{average turnaround time} = \frac{10+9+10+10+13}{5} = 10.4\text{ms}$$

- nonpreemptive SJF:

$$\text{average turnaround time} = \frac{10+9+8+11+13}{5} = 10.2\text{ms}$$

- preemptive SJF:

$$\text{average turnaround time} = \frac{19+1+2+2+5}{5} = 5.8\text{ms}$$

- nonpreemptive priority:

$$\text{average turnaround time} = \frac{10+9+15+15+10}{5} = 11.8\text{ms}$$

- preemptive priority:

$$\text{average turnaround time} = \frac{18+1+10+15+5}{5} = 9.8\text{ms}$$

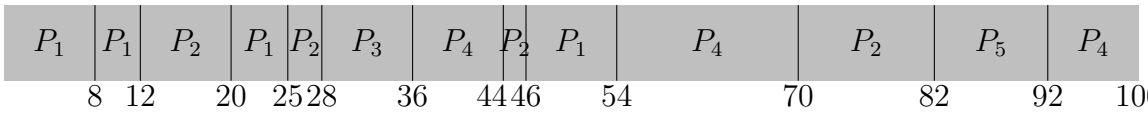
- RR (quantum = 1):

$$\text{average turnaround time} = \frac{38}{5} = 7.6\text{ms}$$

- RR (quantum = 3):

$$\text{average turnaround time} = \frac{42}{5} = 8.4\text{ms}$$

2. (a)



(b) There are 12 context switches for the processes.

(c) The average waiting time is $(8 + 45 + 0 + 32 + 28)/5 = 22.6$ ms. The average turnaround time is $(8 + 64 + 25 + 70 + 46)/5 = 42.6$ ms.

3. (1) If the Shortest job first algorithm is preemptive, then it could result in starvation if the currently executing process has a pretty long CPU burst and there are lots of processes with shorter CPU burst entering the ready queue, the currently executing process would be preempted and may suffer from starvation.

(2) Priority Scheduling algorithm may also result in starvation, since a steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.