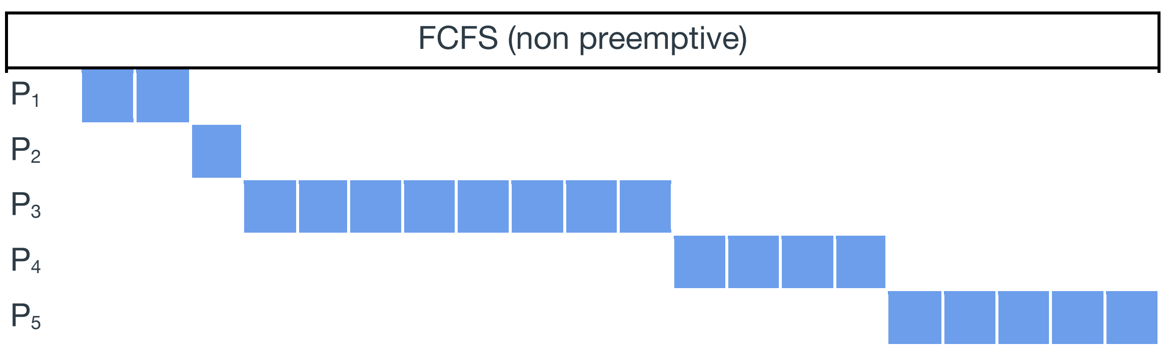
Margaret Connor
CSS 430

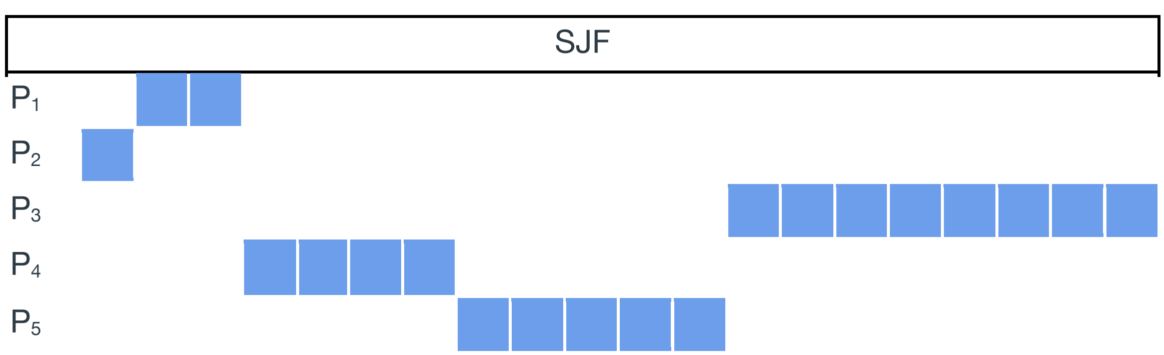
6.16 Consider the following set of processes, with the length of the CPU burst given in milliseconds

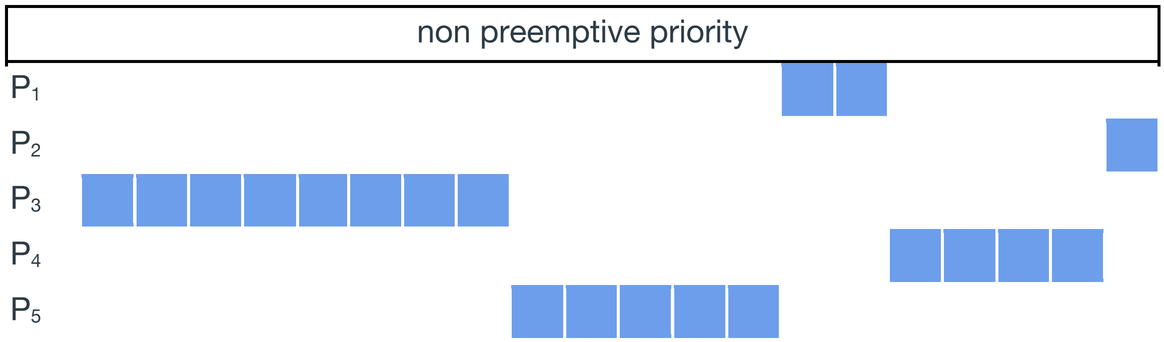
Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

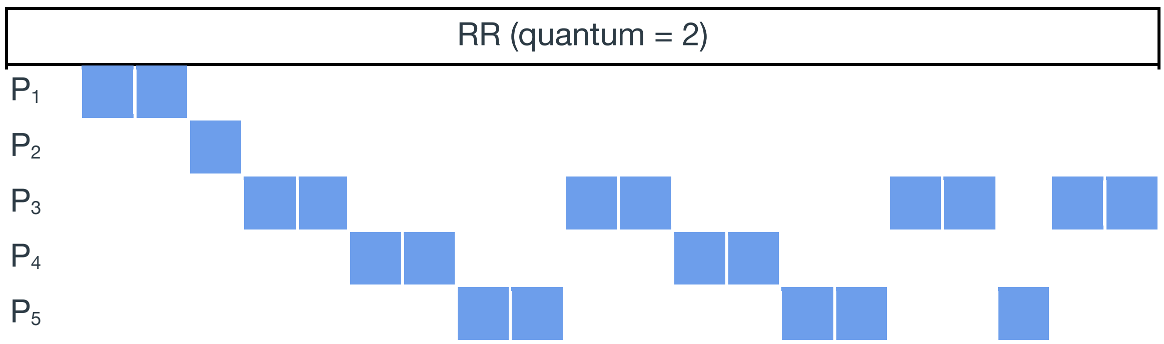
The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0

a. **Draw four Gnatt charts that illustrate the execution of these process using the following scheduling algorithms: FCFS, SJF, non preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).**









b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

	FCFS	**SJF**	**Priority**	**RR**
P_1	2 ms	3 ms	15 ms	2 ms
P_2	3 ms	1 ms	20 ms	3 ms
P_3	11 ms	20 ms	8 ms	20 ms
P_4	15 ms	7 ms	19 ms	13 ms
P_5	20 ms	12 ms	13 ms	8 ms

c. What is the waiting time of each process for each of these scheduling algorithms?

TRT	FCFS	SJF	Priority	RR
P_1	0 ms	1 ms	13 ms	0 ms
P_2	2 ms	0 ms	19 ms	2 ms
P_3	3 ms	12 ms	0 ms	12 ms
P_4	11 ms	3 ms	15 ms	9 ms
P_5	15 ms	7 ms	8 ms	13 ms

d. **Which of the algorithms results in the minimum average waiting time (over all processes)?**

Shortest job first resulted in the shortest average waiting time of 4.6 ms per process.

6.27 Consider the scheduling algorithm in the Solaris operating system for time-sharing threads.

a. **What is the time quantum (in milliseconds) for a thread with priority 15? With priority 40?**

A thread with priority of 15 gets a 160 millisecond quantum.

b. **Assume that a thread with priority 50 has used its entire time quantum without blocking. What new priority will the scheduler assign this thread?**

The new priority of the thread would be be 5.

c. **Assume that a thread with priority 20 blocks for I/O before its time quantum has expired. What new priority will the scheduler assign this thread?**

The new priority of the thread would be 52.

5.10 Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs.

The reason it would not be appropriate is because it allows the user-level programs to prevent context switching from occurring, monopolizing the CPU and circumvent the scheduler.

5.11 Explain why interrupts are not appropriate for implementing synchronization primitives in multiprocessor systems.

Interrupts in multiprocessor system can cause multiple threads to enter a critical section and deadlock one another. Threads on another process could also ignore the critical section on another and access the shared data causing a race condition.