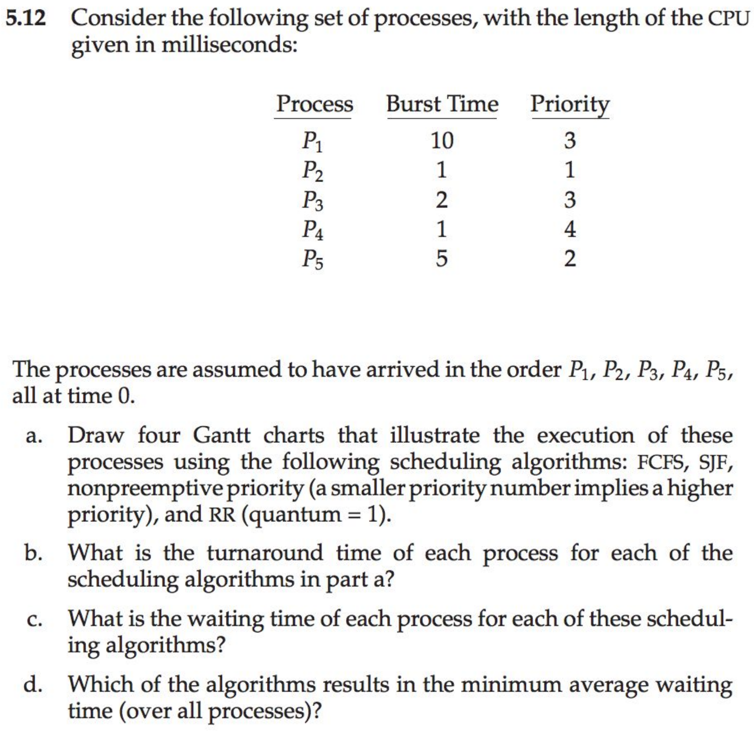


5.12 Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

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

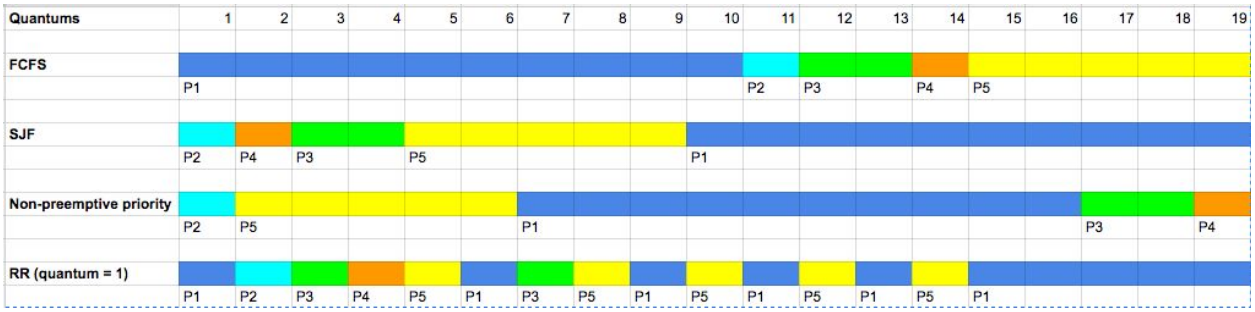
a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

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

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

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

a.



b.

TATs	FCFS	SJF	Non-preemptive priority	RR
P1	10	19	16	19
P2	11	1	1	2
P3	13	4	18	7
P4	14	2	19	4
P5	19	9	6	14

c.

Waiting Times	FCFS	SJF	Non-preemptive priority	RR
P1	0	9	6	9
P2	10	0	0	1
P3	11	2	16	5
P4	13	1	18	3
P5	14	4	1	9

d.

Shortest Job First can be provably shown to always have the shortest avg waiting time, although there is an inherent problem in determining the burst times up front.

6.12 Explain why spinlocks are not appropriate for single-processor systems yet are often used in multiprocessor systems.

A spinlock on a single-processor system is like locking yourself out of the house while nobody else is home, because the process would be spinning on the CPU and waiting for another process to break it out, which of course doesn't exist because it's the only process. A spinlock on a multiprocessor system, however, is like locking yourself out of the house while there's a party going on inside. Just knock on the door and when someone opens it go back in and have another. This is the situation when a process spinlocks on a multiprocessor system such that another processor can change the program state and break the process out of the lock.

6.13 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.

Because you would be giving a malicious process the opportunity to run on the CPU forever and DOS all the other processes.