CS307 Operating System Exercises Spring 2020

Exercise 1: CPU Scheduling

The following rules apply to the exercises:
* We assume that when a preempted process and new coming process need to be put in the ready queue at the same time, we prefer to put the new coming process in front of the preempted process in the ready queue.
* When processes are of same priority, use FCFS to determine the order.

1. Consider the following set of processes, with the arrival times and the length of the CPU-burst times given in milliseconds, and the priorities given so that smaller priority number means a higher priority.

Process	Arrival time	Burst time	Priority
P1	0	10	3
P2	2	2	1
P3	3	2	3
P4	4	1	4
P5	6	5	2

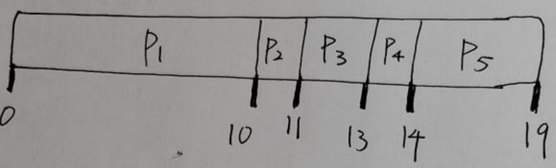
a) Draw Gantt charts illustrating the execution of these processes using the following scheduling algorithm: (1) FCFS, (2) nonpreemptive SJF, (3) preemptive SJF, (4) nonpreemptive priority, (5) preemptive priority, (6) RR (quantum = 1), and (7) RR (quantum = 3). We specify that when a preempted process and new coming process need to be put in the ready queue at the same time, we prefer to put the new coming process in front of the preempted process in the ready queue. We also specify that when processes are of the same priority, use FCFS to determine the order.
b) Calculate the average waiting time when using each of the above scheduling algorithms.
c) Calculate the average turnaround time when using each of the above scheduling algorithms.

Answer:

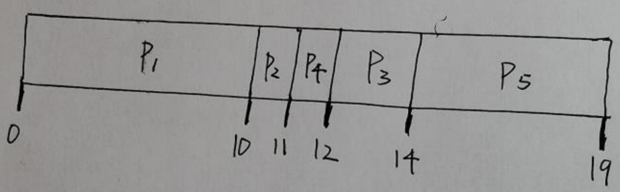
(a)

1

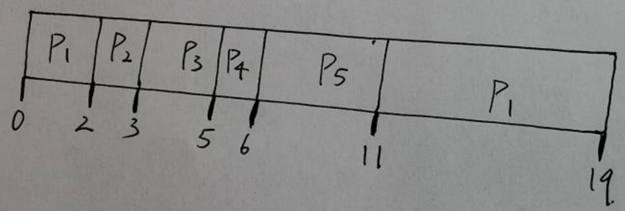
(1) FCFS :



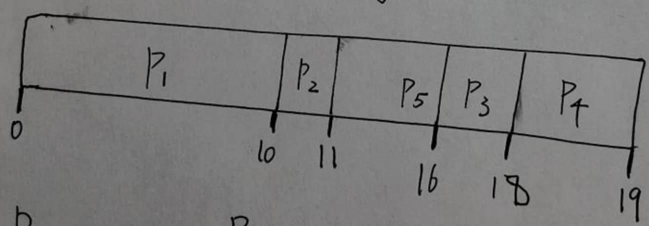
(2) Nonpreemptive SJF :



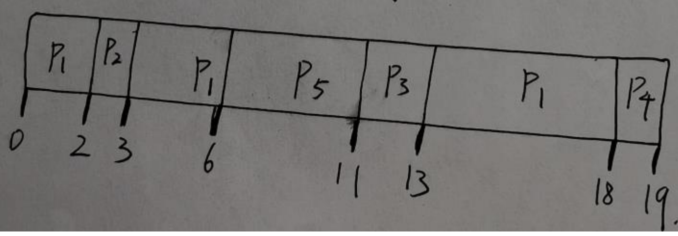
(3) Preemptive SJF :



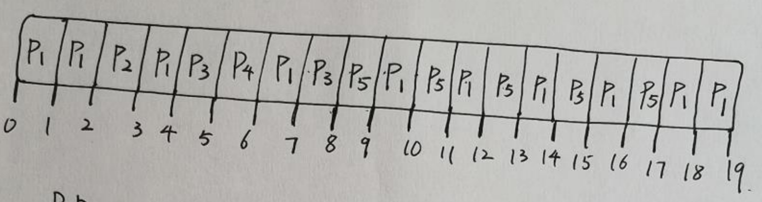
(4) Nonpreemptive Priority :



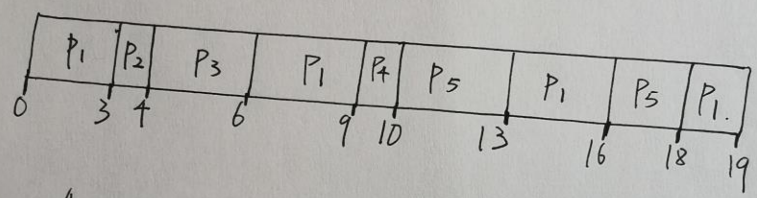
(5) Preemptive Priority :



(6) RR (quantum = 1)



(7) RR (quantum = 3)



(b) Average waiting time:

	P1	P2	P3	P4	P5	Total	Average
FCFS	0	8	8	9	8	33	6.6
NP-SJF	0	8	9	7	8	32	6.4
P-SJF	9	0	0	1	0	10	2
NP-Priority	0	8	13	14	5	40	8
P-Priority	8	0	8	14	0	30	6
RR(q=1)	9	0	3	1	6	19	3.8
RR(q=3)	9	1	1	5	7	23	4.6

(c) Average turnaround time:

	P_1	P_2	P_3	P_4	P_5	Total	Average
FCFS	10	9	10	10	13	52	10.4
NP-SJF	10	9	11	8	13	51	10.2
P-SJF	19	1	2	2	5	29	5.8
NP-Priority	10	9	15	15	10	59	11.8
P-Priority	18	1	10	15	5	49	9.8
RR(q=1)	19	1	5	2	11	38	7.6
RR(q=3)	19	2	3	6	12.	42	8.4.

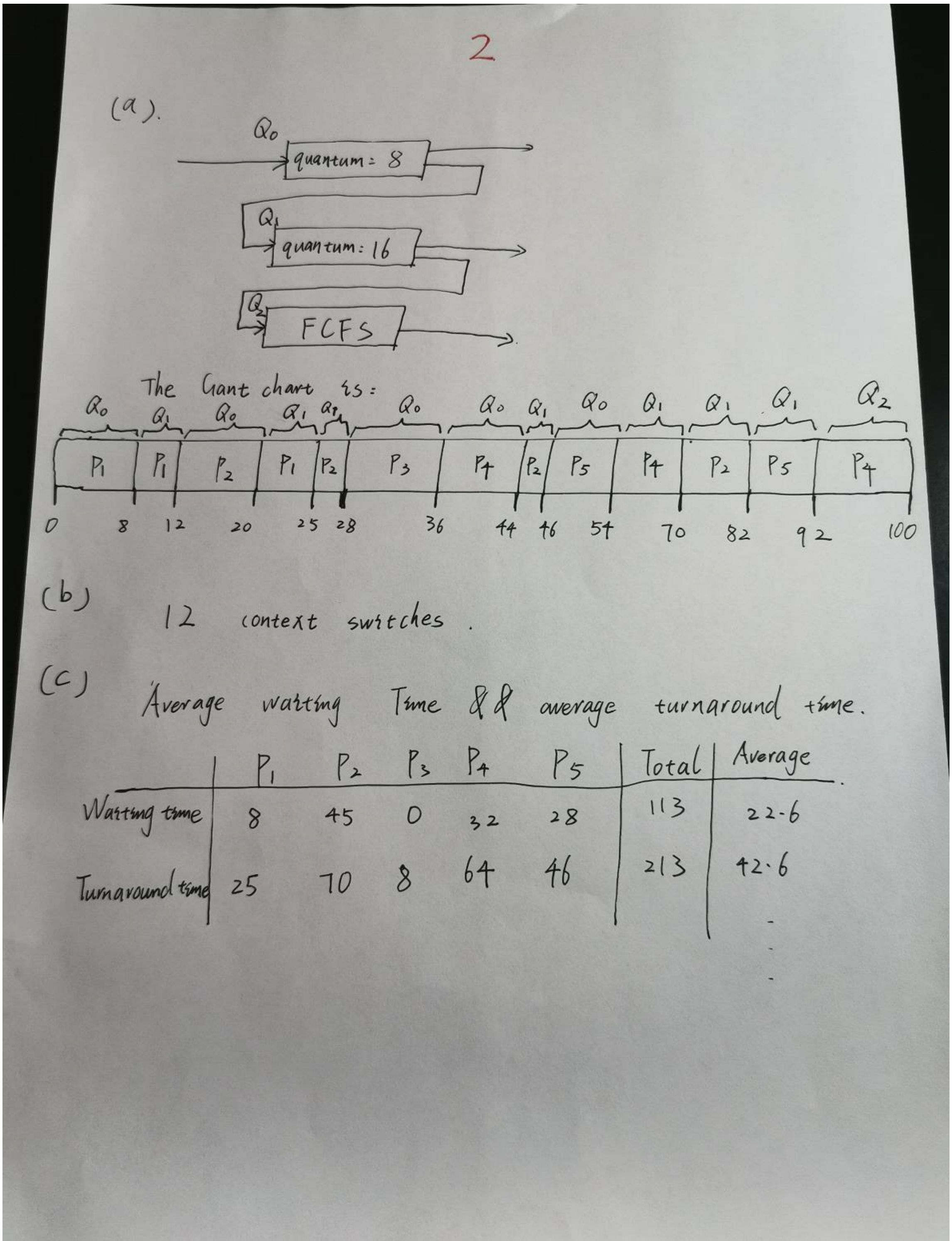
2. Consider the following set of processes, with the arrival times and the length of the CPU-burst times given in milliseconds.

Process	Arrival time	Burst time
P1	0	17
P2	12	25
P3	28	8
P4	36	32
P5	46	18

a) Draw a Gantt chart illustrating the execution of these processes using a Multilevel Feedback Queue Scheduling. Use the same structure of the model as in the lectures: (queue 0: quantum 8), (queue 1: quantum 16), (queue 2: FCFS).

b) Calculate the number of context switches for the processes.

c) Calculate the average waiting time and the average turnaround time for the scheduling.



2

(a).

The Gant chart is:

Q_0	Q_1	Q_0	Q_1	Q_1	Q_0	Q_0	Q_1	Q_0	Q_1	Q_1	Q_1	Q_2
P_1	P_1	P_2	P_1	P_2	P_3	P_4	P_2	P_5	P_4	P_2	P_5	P_4

0 8 12 20 25 28 36 44 46 54 70 82 92 100

(b) 12 context switches.

(c) Average waiting Time && average turnaround time.

	P_1	P_2	P_3	P_4	P_5	Total	Average
Waiting time	8	45	0	32	28	113	22.6
Turnaround time	25	70	8	64	46	213	42.6

3. Which of the following scheduling algorithms could result in starvation, and why?

a. First-come, first-served

b. Shortest job first

c. Round robin

d. Priority

Answer: b d

Reason: In shortest job first case, if a process has a long CPU burst time, it's priority is very low and it might never execute, so this case can cause starvation. In priority scheduling, for the same reason, low priority processes may never execute if every time before it execute and there is a process holds a higher priority lining up in front of it. So this case can cause starvation, too.