CS 5/7343 Fall 2021

Homework 1

Due : 10/13 (Wed) 11:59pm (No extensions)

1. (24 points) You are given the following processes with CPU-burst time, arrival time and priority (lower # means higher priority)

Process	CPU-burst	Arrival time	Priority
P1	10	6	2
P2	11	0	4
P3	4	3	2
P4	8	4	1
P5	9	1	2

For each of the following scheduling algorithm, show (using the diagram as in the slides), how the process are being executed. Also calculate the average wait time.

a. Shortest job first (non-preemptive)
b. Shortest remaining job first (preemptive)
c. Priority-based (preemptive, with round robin on process with same priority, quantum = 3, quantum only applied to processes that is involved in round robin).
d. Priority-based (preemptive, with round robin on process with same priority, quantum = 5 quantum only applied to processes that is involved in round robin).

A couple of other details:

- If you need to preempt a job, only preempt when the new job has a strictly better criteria (priority, wait time etc.) Do not preempt if the criteria are the same.
- When a job arrives at the first time, it is put into the appropriate position of the queue based on the criteria (priority, remaining time etc.). If there is a tie, it will be put after all the jobs that is already in the queue.
- For round robin, if a process is to be swap back to the end of the queue, and another process arrived at the same time, the newly arrived process should enter the queue first. Also, if multiple process that have the same priority can start, pick the one that is currently closer to the front of the queue.

You should present your output in the table below (the first three steps of ***case (a***) is done for you):

Time	Current Process Being Run	Order of the processes in the wait queue
0	P2	--
1	P2	{P5}
3	P2	{P3, P5}

You should add an entry whenever

- A process just arrived
- A process is swapped out (or finished execution) and another process is swapped in.

You do not need an entry when the last process is finished.

2. (18 points) Consider real time process for the following 3 process with the period and running time below:

Process	Period	Running time
1	50	20
2	60	20
3	25	5

   a. What is the overall CPU utilization of the three processes combined?
   b. Will Rate Monotonic Scheduling be able to schedule the three process? Prove it one way or the other.
   c. Repeat part (b) with Earliest Deadline Scheduling

3. (10 points) Consider using test_and_set() to solve the critical section problem (slide 26, process synchronization). Now assume that test_and_set() operation is not atomic. Explain why it will not solve the critical section problem.

4. (12 points) Consider Peterson's solution for the critical section. For each of the following modification (they are separate – not one on top of the other), argue whether the modified solution still solve the critical section problem. Explain your answer:

   a. Add the line (turn = i) after the while loop is exited but before entering the critical section
   b. Change line 2 inside the big while loop from (turn = j) to (turn = i)