King Saud University
College of Computer and Information Sciences
Computer Science Department

College of Computer & Information Sciences
Computer Science Department

CSC227: Operating Systems
Course Project – 2nd Semester 1445
Due Date: Thursday April 25, 2024

I. Problem Statement

This project will involve implementing a Multilevel Queue (MLQ) scheduling algorithm with two queues, queue numbered as Q1 and queue numbered as Q2 with specific assigned priority for each queue (Q1 has a higher priority than Q2). Each process must occupy the respective queue with specific priority range according to its priority (1 or 2). The scheduler follows Round-Robin (RR) scheduling algorithm with time quantum of (3 ms) for processes in Q1 and non-preemptive Shortest-Job-First (SJF) algorithm for processes in Q2. The multilevel queue-scheduling algorithm is implemented as a fixed preemptive priority scheduling for each and every queue. It is assumed that processes arrive at different times, so your scheduler must support higher-priority processes preempting processes with lower priorities when process arrives at high priority queue.

Note: If a new process arrives at the same time a process releases the CPU, then the new process will be added to the ready queue first.

Your program should show the following menu:

1. Enter process' information.
2. Report detailed information about each process and different scheduling criteria.
3. Exit the program.

II. Project Description

In this project, the students will write a program to schedule processes as a fixed preemptive priority scheduling for each and every queue in the Multilevel Queue (MLQ) scheduling algorithm. Your program should contain two classes, the first class is used to represent the Process Control Block (PCB), and the second class is used as a driver class, which contains the main method (the implementation of each one of them depends on the students). Moreover, your program should provide a report about each process, which includes [process ID, priority, arrival time, CPU burst time, starting and termination time, turnaround time, waiting time, and the response time]. In addition, the report should contain information about different scheduling criteria, which include [the average turnaround time, the average waiting time, and the average response time]. The program should behave as follows:

1. The program will create two arrays of type PCB to represent Q1 and Q2.
2. The program will prompt the user to enter the number of processes (P).
3. The program will prompt the user to enter the priority, arrival time, and CPU burst of each process.
4. The program will create a process (object) of type PCB for each process and initialize its attributes:
    - Process ID: In the form of P#, where # represents the process number. (Assume the numbering starts from 1)
    - Priority of a process: priorities range from 1 to 2, where a lower numeric value indicates a higher relative priority.
    - Arrival time: The time at which the process enters into the ready queue.

- CPU burst: The amount of time a process needs to execute.
- Start and termination time: The start time of process execution and the completion time.
- Turn around time: The amount of time to execute a process.
- Waiting time: The amount of time a process has been waiting in the ready queue.
- Response time: The amount of time it takes from when a request was submitted until the first response is produced.

5. The program will add the process to either Q1 or Q2, depending on its priority.
6. The program will schedule processes execution in the CPU and output on the console a report of each process in system with the following information, and then will write them to a text file:
   - Show the scheduling order of the processes. Example: [P1 | P2 | P1 | P3].
   - Process ID.
   - Priority.
   - Arrival time.
   - CPU burst.
   - Start and termination time.
   - Turnaround time.
   - Waiting time.
   - Response time.
   - Average [Turnaround time, Waiting time, and Response time] for all processes in the system.
7. The program will display a menu and prompt the user to input her choice [1, 2, or 3] based on the list mentioned in the *problem statement* section above.
8. If the user selects option 1, the user should be requested to enter the total number of processes in the system and then enters each processes' information, namely, the process priority, the arrival time and the CPU burst time.
9. If the user selects option 2, your program will display the scheduling order of the processes, detailed information about each process in the system as described above, and different scheduling criteria on the console, and then writes these information to the output file (*Report.txt*).
10. If the user selects option 3, you should exit the program.

III. Deliverables

The Team Leader is required to submit a single zip folder on the behalf of the group containing the following.

1. Program code in softcopy**. Java** language should be used.
2. A Read Me file in PDF containing the following:
   a. Student names and IDs highlighting Team Leader.
   b. Table showing task distribution. *There should be a clear and practical distribution of tasks.*
   c. Screen shots (using clear, descriptive captions) showing sample input/output so that:
      i. The RR condition is demonstrated with different scenarios (i.e., some processes have less/equal/more CPU burst time than the specified time quantum (3ms))
      ii. The non-preemptive Shortest-Job-First (SJF) scheduling is demonstrated.
      iii. Preemption is demonstrated (a newly arriving process has a higher priority than the currently executing process)

   In addition, a screenshot of the report (*Report.txt*) should be included.

d. Student peer evaluation form (see Team Work Evaluation table below).

IV. Evaluation Rubric

Evaluation rubric is divided into two parts: First part evaluates Team Work and second part addresses the functional requirements. The code in grade assignment is shown below.

Code	
Fully satisfied	1
Partially satisfied	0.5
Not satisfied	0

1. Team Work Evaluation

Part 1: Team Work			
Criteria	**Student 1**	**Student 2**	**Student 3**
Work division: Contributed equally to the work			
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA			
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions			
Time management: Attending on time, being ready to start the demo, good time management in discussion and demo.			
Total/4	0	0	0

2. Functional Requirements

Part 2: Functional Requirements		
	Criteria	**Evaluation**
Implementation	Overall quality of the code implementation (organization, clearness, design,…)	1
	Create two arrays to represent Q1 and Q2.	1
	Initialize the process PCB (object) attributes and assign it to Q1 or Q2 correctly.	1
	Logic for scheduling processes as a preemptive priority scheduling in the multilevel queue is correct and complete.	3
	Logic for scheduling processes using RR for processes within Q1.	3
	Logic for scheduling processes using SJF for processes within Q2.	3
	Logic for calculating the start and termination time is correct and complete.	1
	Logic for calculating the turnaround time is correct and complete.	1
	Logic for calculating the waiting time is correct and complete.	1
	Logic for calculating the response time is correct and complete.	1
	Logic for computing the average turn around, waiting, and response time is correct and complete.	1
	Program displays the desired output on the console correctly and completely.	1
	Program displays the desired output (Report.txt) correctly and completely.	1
Deliverables	1. Program code as in III.1.	1
	2. A Read Me file in PDF as in III.2.	1
	Total	21