Second Semester – 2023/2024
Operating Systems (ITCS323/325) - Project
Due Date: Tuesday, 21/5/2024, 11:59 pm (via the blackboard)

- The purpose of this project is to apply the theoretical concepts (threads, synchronization, and scheduling) taken in the course into real application.
- The project will contribute 10 points towards your total grades.
- This is a group project. Each group consists of 4-5 students **ONLY.**
- Remember, each person is expected to contribute equally on the class project and each team is expected to do their own work (**do not collaborate with others outside the team and do not use any external resources**), otherwise each person involved will be subject to the University Dishonesty Policy. In addition, ZERO will be given for the work.

Choose one of the following projects:

1. CPU Priority Scheduling with Round Robin:

Write a Java program to implement a priority scheduling algorithm with round robin. Your program should first prompt the user to **input** quantum time (q) followed by a list of *process ID*, *arrival time*, *burst time*, and *priority* for each process to be run on the CPU (2 marks). The list is terminated by 0 0 0 0 for the process ID, arrival time, burst time, and priority. The program **output** should *draw a Gantt chart (as text, e.g., 0-p1-3-p3-5-p5-9-……)* that shows the scheduling order of the processes using the priority with RR scheduling algorithms (5 marks). Also print the *turnaround time*, *response time*, and *waiting time* for each process along with their *average* for all processes (3 marks). Make sure to *display helpful messages* to the user for input and output.

2. Shortest Job First with Round Robin:

Write a Java program to implement a shortest job first scheduling algorithm with round robin. Your program should first prompt the user to **input** a list of *process ID*, *arrival time*, and *burst time* (in msec) for each process to be run on the CPU (2 marks). The list is terminated by 0 0 0 for the process ID,

arrival time, and burst time. The program **output** should *draw a Gantt chart (as text, e.g., 0-p1-3-p3-5-p5-9-……)* that shows the scheduling order of the processes using SJF with RR scheduling algorithms (5 marks). Also print the *turnaround*, *response*, and *waiting time* for each process along with their average for all processes (3 marks). Make sure to *display helpful messages* to the user for input and output.

IMPORTANT Submission and Discussion Instructions:

- The project will be submitted on Tuesday 21/5/2024, 11:59 pm via the blackboard.
- The project discussion schedule (date and time) will be announced through the blackboard for each project team.
- The project discussion will not exceed 10 minutes. I will ask questions for each student in a group.
- At the beginning of your code, please add the students' names and IDs as comments.
- **Please submit ONE zipped folder ONLY which contains the following two files:**
  - **A pdf report that contains the following:**
    - **First Page: the name of your selected CPU scheduling algorithm and students' names and IDs.**
    - **Second Page or more: copy your java code (as text)**
    - **Third Page: input and output screen for two examples (as images)**
  - **Java file.**
- **Name the zipped folder as the student ID of the team leader.**
- **ONLY the team leader should submit the project on the blackboard.**
- **How to get full mark in the project? I will run your code in some examples I have, then your code should show the correct answers of all requested results by 100%. Also, your code should not have any copy.**