

# ***Operating System***

## ***Fall 2023 – Assignment 2***

---

### **Instructions:**

- It's a group assignment and can be done in a group of two.
- Copied assignments will be marked negatively.
- Late submissions will not be entertained in any case.
- Online resources can be used for your understanding only but basing your whole assignment on it can result in losing all your marks, with a penalty on other assessments.

### **Problem Statement**

In this assignment, you will implement and simulate different scheduling algorithms for a set of processes. You will be given a file containing process information, including Process IDs (p1, p2, p3 ...), burst times, arrival times, and priorities. Your task is to create a console-based simulation for the following scheduling algorithms:

- First Come First Serve
- Shortest Process Next (SPN)
- Shortest Remaining Time
- Highest Response Ratio Next
- Round Robin ( $q = 1$ )
- Round Robin ( $q = 2$ )
- Multi-level Feedback Queue (MLFQ) with  $RQ=1$
- Multi-level Feedback Queue (MLFQ) with  $RQ=2$ <sup>1</sup>

For each scheduling algorithm, your simulation should include the following components:

1. Initialize the ready queue and any necessary data structures.
2. Simulate the execution of processes based on the scheduling algorithm.
3. Display a Gantt chart to visualize the sequence of processes running on the CPU.
4. Show when each process gets CPU time (running), when it goes to the ready queue, and when it exits.

### **You can use Process Timing Diagram to explain better**

After simulating each scheduling algorithm, calculate the following performance metrics for each algorithm:

1. Turnaround Time: The total time taken by a process to complete its execution, from arrival to exit.
2. Response Time: The time it takes for a process to receive the CPU for the first time.
3. Waiting Time: The total time a process spends waiting in the ready queue.
4. Your program should provide clear and informative output, showing the progress of each process through the scheduling algorithm.

**Additional Guidelines:**

- You can choose an appropriate programming language for this assignment.
- Ensure that your program reads the process data from a file and provides user-friendly command-line interaction.
- Clearly document your code, including explanations of your scheduling algorithms and any data structures used.
- Test your program with different sets of processes to demonstrate the effectiveness of each scheduling algorithm.

**Submission:**

Submit your source code and a report that includes:

- A description of each scheduling algorithm you implemented.
- Code snippets or pseudo-code for critical parts of your implementation.
- Screenshots or diagrams of Gantt charts generated during the simulation.
- Results and analysis of turnaround time, response time, and waiting time for each algorithm.