CSC227 Operating System
Course Project (100 points)
Due Date: 30 November 2025 @ 11:59 pm (via Blackboard)

Objective:

The aim of this project is to write a java program that simulates the CPU Scheduler and creates multiple threads to accomplish specific tasks in the simulation. This is a group project. Each group can have up to 3 students.

System Description:

Consider a simple system with a single CPU. The system will serve several jobs that all are arrived at time 0. You are required to simulate three different scheduling algorithms:

1) Shortest Job First (SJF) (Non-Preemptive)
2) Round-Robin (RR) – quantum = 7ms
3) Priority scheduling (1->Lowest priority, 128->Highest Priority)

Each job has a PCB that contains all required information to identify the job such as id, state, burst time, required memory, required statistics which include turnaround time for each job and waiting time for each job. You can add other fields to PCB to help you in programming.

The program will read process information from a file (job.txt) - this file will be provided by the instructor. File reading should be performed in an independent thread that creates the PCBs and put them in the job queue. Loading the jobs to ready queue should be performed in an independent thread that continuously checks the available space in memory to load the next job from the job queue to the ready queue. You must make sure that jobs can be loaded to ready queue only if enough space for the job is available in memory.

The main thread will perform the scheduling algorithms. Your program should let the user to choose the scheduling algorithm to be applied. Your program should simulate the use of system calls. You are supposed to propose a set of system calls for process control, information maintenance, and memory management.

For the Priority scheduling, your program should identify starvation of a process, if the process waited more than the degree of multiprogramming at the time of the process acceptance to the ready queue, and implement aging (for priority scheduling) to resolve the issue.

Structure of the input file:

A sample input file of three jobs is given as follows (Process ID: burst time in ms: priority; Memory required in MB,): [Begin of job.txt]

1:25:4;500
2:13:3;700
3:20:3;100

[End of job.txt]

You can assume that

1) There are no more than 30 jobs in the input file (job.txt).
2) Processes arrive in the order they are read from the file.
4) Main memory is limited to 2048 MB
5) Context switching time is zero.
6) **Your application must use threads (minimum two).**

Your Output:

You should compare the *average waiting times* and the *average turnaround times* of all jobs for each scheduling algorithm.

Output the details of each algorithm's execution. You need to show which jobs are selected at what times as well as their starting and stopping burst values, indicate which process suffered from starvation in case of the priority scheduling. You can choose your display format, but it is recommended that you display the results in terms of a Gantt chart. **Turn in**:

1) Soft copy of the program with proper documentation.
2) Soft copy of the report (at least two pages) of this programming assignment including:
   a. software and hardware tools you used
   b. description of system calls proposed, including and not limited to, arguments, expected output, and the functionality of the proposed system call.
   c. strength and weakness of your program
   d. How many threads your application created? does multithreading speed up the performance of your application? Explain why.
   e. do you prefer to read Gantt chart as an output of a processor schedule? or you like the other methods such as tables? Explain why.
   f. if you need to simulate the whole operating system (instead of just the CPU Scheduler here), explain how you can modify your program to do that.

Presentation Day:

1) Print and bring the project cover sheet on the day of presentation (You can download the project cover sheet from the Blackboard).
2) Your program will be evaluated based on a new input file on the day of presentation.
3) You will be asked some questions regarding your experience on this project.