

CSCE 5640: Operating System Design

Project Proposal

Name: Kishan Kumar Zalavadia

EUID: 11685261

1. Overview and objective(s) of the project.

Scheduling Algorithms:

- a. This project involves implementing several different process scheduling algorithms. The scheduler will be assigned a predefined set of tasks and will schedule the tasks based on the selected scheduling algorithm. Each task is assigned a priority and CPU burst.
- b. Implement the following scheduling algorithms:
 - i. First-come, first-served (FCFS), which schedules tasks in the order in which they request the CPU.
 - ii. Shortest-job-first (SJF), which schedules tasks in order of the length of the tasks' next CPU burst.
 - iii. Priority scheduling, which schedules tasks based on priority.
 - iv. Round-robin (RR) scheduling, where each task is run for a time quantum (or for the remainder of its CPU burst).
 - v. Priority with round-robin, which schedules tasks in order of priority and uses round-robin scheduling for tasks with equal priority.
- c. Priorities range from 1 to 15, where a higher numeric value indicates a higher relative priority. For round-robin scheduling, the length of a time quantum is 10 milliseconds.
- d. Implementation:
 - i. Each algorithm should take a schedule of tasks from external files.
 - ii. The schedule of tasks has the form **[task name] [priority] [CPU burst]**, with the following example format:

T1, 4, 20

T2, 2, 25

T3, 3, 25

T4, 3, 15

T5, 10, 10

- iii. Thus, task T1 has priority 4 and a CPU burst of 20 milliseconds, and so forth. It is assumed that all tasks arrive at the same time, so your scheduler algorithms do not have to support higher-priority processes preempting processes with lower priorities.
- iv. Example of running executable in C/C++ for FCFS: `./fcfs schedule.txt`
 - Executing in Java (if you use Java): `java FCFC schedule.txt`
- e. Create test cases and compare the scheduling algorithms to find the best performing algorithms based on metrics such as average waiting time, turnaround, response time etc.
 - i. You should create multiple test cases with the same and different number of processes, their priorities and burst times in separate files for testing.
 - ii. For example, create at least 5 schedule files with 5 processes with their priorities and burst times, create at least 5 schedule files with 10 processes with their priorities and burst times, create at least 5 schedule files with 15 processes with their priorities and burst times, etc.
 - iii. You may randomly generate priorities and burst times for the test schedule files.
 - iv. Analyze your experimental results and draw conclusion on which scheduling algorithm performs better in your overall experiments.
- f. See more in Chapter 5 (CPU Scheduling) -> Programming Projects -> Scheduling Algorithms

- The main goal of the project is to compare various scheduling algorithms.
- The objectives are implementing the scheduling algorithms (FSFS, SJF, Priority, RR, Priority with RR).
- Read the tasks from the external file.
- Create multiple test cases to test the performance of the scheduling algorithms.
- To analyze and compare the algorithms using metrics like average waiting time, turnaround time, and response time.
- Draw the conclusion based on the experiments about the effectiveness of the scheduling algorithms.

2. Team size and team members.

Team size: 1

Team Member: Kishan Kumar Zalavadia (11685261)

3. Project Plan

a. Task divisions for the team members.

I am handling everything by myself.

- Create a Java project to read a file.
- Implement FCFS.
- Implement SJF.

- Implement Priority Scheduling.
- Implement RR.
- Implement Priority with RR.
- Creating test cases and comparing the algorithms.
- Creating project documentation.

b. Due date for subtasks.

Oct 25, 2024: FCFS

Oct 26, 2024: SJF

Oct 27, 2024: Priority Scheduling

Oct 28, 2024: RR

Oct 29, 2024: Priority with RR

Oct 30, 2024: Create test cases and compare the scheduling algorithms

Nov 01, 2024: Analyze the results

Nov 02, 2024: Creating project documentation.

4. Experimental environment

a. Programming language for implementation.

This project may be implemented in Java.

b. Operating system to test the project.

This project will be developed on a Linux CSE machine.

It can be tested on any OS.

c. Test cases.

I will create seven schedule files with five processes and their priorities and burst times, seven schedule files with ten processes and their priorities and burst times, and seven schedule files with fifteen processes and their priorities and burst times.

I may create a script that generates all these files with random values and test it. I may use average waiting time, turnaround time, and response time to compare the efficiency of the algorithms.
