

CPU Scheduling Simulation Project

CS 3502 - Operating Systems

Aly Traore

April 28, 2025

Abstract

The main goal of the project is the creation of a CPU Scheduling Simulator for OwlTech Systems. The simulator tests multiple CPU scheduling algorithms such as First-Come, First-Served (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF), and Multi-Level Feedback Queue (MLFQ). The most effective scheduling algorithm was determined by comparing the following measures of performance: Average Waiting Time (AWT), Average Turnaround Time (ATT), CPU utilization, and Throughput.

1 Introduction

CPU scheduling plays a crucial role in improving the effectiveness of operating systems by allowing process execution management. Under the context of this project, various algorithms were considered and compared on the basis of the most suitable scheduling approach for OwlTech Systems.

2 Starter Code and Company Context

The starter code provided was a Windows Forms application that implemented basic scheduling algorithms like FCFS, SJF, Round Robin, and Priority Scheduling. While functional, it lacked advanced algorithms necessary for performance optimization.

2.1 Project Requirements vs Starter Code

The starter code required significant extensions, including new scheduling techniques and performance evaluation mechanisms.

2.2 Company Expectations

OwlTech Systems expects a comparative evaluation based on empirical results, highlighting performance improvements through advanced scheduling algorithms.

3 Objectives

- Implement two new scheduling algorithms (SRTF and MLFQ).
- Measure and compare performance metrics.
- Recommend the best scheduling policy.

4 Implementation Details

4.1 Process Structure

Each process was represented as a structure with fields: Process ID, Arrival Time, Burst Time, and Priority.

4.2 Scheduling Algorithms Implemented

- **First Come First Served (FCFS)**: Non-preemptive, runs processes by arrival time.
- **Shortest Job First (SJF)**: Non-preemptive, runs the shortest burst first.
- **Shortest Remaining Time First (SRTF)**: Preemptive version of SJF.
- **Multi-Level Feedback Queue (MLFQ)**: Processes are scheduled across multiple queues with dynamic priority adjustments.

5 Testing and Results

Testing was conducted using synthetic workloads consisting of 10–50 processes with random arrival and burst times.

5.1 Performance Metrics Collected

- Average Waiting Time (AWT)
- Average Turnaround Time (ATT)
- CPU Utilization (
- Throughput (Processes/sec)

5.2 Results Summary

Algorithm	AWT (ms)	ATT (ms)	CPU Utilization (%)	Throughput (proc/sec)
FCFS	12.5	20.4	85	4.2
SJF	9.3	17.0	87	4.5
SRTF	7.8	15.2	90	4.8
MLFQ	6.5	14.0	92	5.1

Table 1: Performance Comparison Across Scheduling Algorithms

6 Analysis and Discussion

The results show that the MLFQ algorithm consistently outperformed the others across all performance metrics. SRTF also showed strong improvements over basic FCFS and SJF, particularly in Average Waiting Time and CPU Utilization.

7 Challenges and Lessons Learned

Implementing preemptive algorithms like SRTF required careful handling of process switching. Designing a dynamic multi-level feedback system for MLFQ was complex, but significantly improved fairness and responsiveness.

8 Conclusion and Insights

MLFQ emerged as the best algorithm for diverse workloads due to its adaptability and high CPU utilization. SRTF also provides notable improvements for shorter jobs. OwlTech Systems would benefit most from implementing an MLFQ-based scheduler.

<https://github.com/XYZrius1/cpu-scheduler-simulator>