

Project 3. CPU Scheduling Simulator (SJF and RR)

Overview

This project involves the development of a CPU Scheduling Simulator in C, focusing on two key scheduling algorithms: Shortest Job First (SJF) and Round Robin (RR). The simulator aims to demonstrate the functioning and efficiency of these algorithms in managing and scheduling processes in a simulated environment.

Project Objective

- To implement and understand the Shortest Job First (SJF) and Round Robin (RR) CPU scheduling algorithms in a **preemptive** format.
- To analyze and compare the performance of these algorithms in terms of process waiting time and turnaround time.

Task Description

1. **Implementing SJF and RR Algorithms:**
 - Develop functions to simulate the SJF and RR scheduling algorithms.
 - Ensure that SJF handles processes based on burst time, and RR uses time quantum for process scheduling.
1. **Process Management:**
 - Create a structure to represent a process with attributes like process ID, arrive time, burst time, waiting time, and turnaround time.
 - Manage the processes in a queue or list for efficient scheduling.
2. **Performance Metrics:**
 - Calculate key metrics such as average waiting time and average turnaround time for each algorithm.
 - In the RR algorithm, you can customize the quantum, but in testing, we will use different data to verify the effectiveness of the algorithm you provide.
3. **Terminal Output:**
 - Display the scheduling order, waiting time, and turnaround time for each process on the terminal.

Evaluation Tests

1. **Algorithms:** Implement the basic logic for SJF and RR as discussed on lecture 6.2
2. **Output in terminal:** Print out the process ID, arrive time, Brust time, waiting time and turnaround time on the terminal.
3. **Calculate the execution order:** Print out the execution order on the terminal after implement provided algorithm.
4. **Makefile:** The programs (SJF & RR) should be compiled by make.

Submission Requirements

1. **Source Code:** Submit the complete C source code files implementing both SJF and RR algorithms, and Make File to GitHub.
2. **Execution Results:** Provide screenshots to show the output of your programs to GitHub.
3. **Submit GitHub link to Canvas.**