CPU-Scheduler Report

Project Overview
In this project, I implemented various CPU scheduling algorithms and evaluated them based on parameters such as Finish Time, Turnaround Time, and Wait Time to determine their suitability for a given set of inputs. The parameters considered in the program include the arrival time, burst time, priority of a process, and the timescale. The implemented algorithms are First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin (RR).

FCFS Algorithm
In the FCFS algorithm, processes are sorted by their arrival time. The process that arrives first is executed first. If two processes have the same arrival time, the one that appears first in the input is executed first. Finish time is calculated during execution, and Turnaround Time and Wait Time are derived from it. A notable issue with FCFS is the potential for process starvation when a large process arrives early.

SJF Algorithm
For the SJF algorithm, processes are first sorted by their arrival time. At each time instant, the shortest job ready for execution is selected and executed. This process continues by selecting the shortest job among available processes. Finish time, Turnaround Time, and Wait Time are calculated similarly to FCFS. This implementation uses a non-preemptive approach and can be improved by implementing a preemptive version for better results.

Priority Scheduling
Processes are sorted by arrival time, and at any given point, the process with the highest priority is executed first. If multiple processes are ready, the one with the highest priority among them is chosen. This implementation considers higher numerical values as higher priority, but user input can be modified to define priority levels. A preemptive version can be implemented for better results.

Round-Robin
In the Round-Robin algorithm, processes are sorted by arrival time and added to a ready queue. Each process in the queue is executed for a time slice, after which the next process in the queue is executed. Any new processes ready for execution are added to the queue. This method reduces response time but increases the finish time for all processes.

Running the Project
- Clone the repository from GitHub and run the provided code with the given input file.

Learning Takeaways
- Gained a comprehensive understanding of various scheduling algorithms, their advantages, and disadvantages.
- Learned the importance of scheduling terms and their impact on selecting the best scheduler.

- Recognized the critical role of schedulers in ensuring efficient execution of tasks in an operating system.
- Enhanced skills in writing clean and efficient C++ code for specific algorithms.
- Learned to integrate front-end and back-end systems to display backend data to users on the front end.

References

- https://www.youtube.com/playlistlist=PLBlnK6fEyqRitWSE_AyyySWfhRgyA-rHk
- https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/Linux/Linux%20Kernel%20Development,%203rd%20Edition.pdf
- https://www.researchgate.net/publication/49619229_An_Improved_Round_Robin_Schedduling_Algorithm_for_CPU_Scheduling