Project report

CPU Scheduling Algorithms Implementation

NAME- AMAN KUMAR
ENROLLMENT NO. – 22116012

DEPT & YEAR - ECE III

JUNE 2024

Introduction

This project implements four classic CPU scheduling algorithms: First-Come, First-Served (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin. The implementation is done in C++ and aims to simulate the scheduling process, calculating and comparing the average waiting and turnaround times for a set of processes.

Code Structure

The code is structured into several functions, each representing a different scheduling algorithm. Below is a brief overview of each function and its purpose:

Data Structures

Process: A structure representing a process with the following attributes:

- `id`: Process ID.
- `Arrival_time`: Time at which the process arrives in the ready queue.
- `burst_time`: Total time required by the process for execution.
- `priority`: Priority of the process (used in Priority Scheduling).
- `remaining_time`: Remaining burst time for the process (used in Round Robin).
- `start_time`: Array to store start times for each quantum (used in Round Robin).
- `wait_time`: Waiting time of the process.
- `finish_time`: Time at which the process finishes execution.
- `turnaround_time`: Turnaround time of the process.

Functions

- **FCFS**: Implements the First-Come, First-Served scheduling algorithm.
- **SJF**: Implements the Shortest Job First scheduling algorithm.
- **Priority Scheduling**: Implements the Priority Scheduling algorithm.
- **Round Robin**: Implements the Round Robin scheduling algorithm.

Each function calculates the average waiting time and turnaround time for the given set of processes.

Main Function

The `main` function initializes the processes, reads input data, and calls each scheduling algorithm function. It also prints the average waiting and turnaround times for each algorithm.

How to Run

1. Compile the code using a C++ compiler.
2. Run the executable and input the number of processes followed by the arrival times, burst times, priorities, and the quantum for Round Robin scheduling.

Conclusion

This project provides a comprehensive simulation of four fundamental CPU scheduling algorithms. By comparing the average waiting and turnaround times, one can determine the most efficient algorithm for a given set of processes and constraints.