Operating System Lab
Course Code: CSE-3632

Process Scheduler Simulator

Team Members

Jawadul Karim
ID: C221010

Abu Tanvir Hasan Tanmoy
ID: C221001

Md. Shariful Islam Junaed
ID: C221033

Submitted To
Zainal Abedin

Department of Computer Science and Engineering (CSE)
International Islamic University Chittagong, Kumira

Abstract

Process scheduling is a fundamental concept in operating system design, aiming to optimize CPU utilization and ensure fair process execution. This project implements and analyzes scheduling algorithms including First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR), and Priority Scheduling. Metrics such as average waiting time and turnaround time are calculated to evaluate algorithm performance. The project features an intuitive graphical user interface (GUI) for inputting process data, selecting algorithms, and visualizing results through Gantt charts.

Introduction

Efficient process scheduling is crucial for modern operating systems to ensure optimal CPU utilization and user satisfaction. This project provides a hands-on approach to understanding scheduling algorithms by implementing a Process Scheduler Simulator with a user-friendly GUI. Users can input processes, choose scheduling algorithms, and view results in real-time.

Features of the Project

- Implementation of popular scheduling algorithms: FCFS, SJF, RR, and Priority Scheduling.
- Calculation of average waiting time and turnaround time for performance evaluation.
- Visualization of results using Gantt charts.
- A graphical user interface designed with Tkinter for easy interaction.

Implementation Details

The project consists of the following modules:

1. **Scheduler Module:** Implements the scheduling algorithms and computes metrics.
2. **GUI Module:** Provides an interface for users to input data, select algorithms, and view results.
3. **Utility Module:** Prints metrics and plots Gantt charts for visualization.

Scheduling Algorithms Implemented

First Come First Serve (FCFS)

Description: Processes are executed in the order they arrive.
Advantages: Simple to implement.
Disadvantages: Causes convoy effect (long processes delay shorter ones).

Shortest Job First (SJF)

Description: Processes with the shortest burst time are executed first.
Advantages: Minimizes average waiting time.
Disadvantages: May lead to starvation of long processes.

Round Robin (RR)

Description: Each process gets a fixed time slice (quantum) to execute in a cyclic manner.
Advantages: Provides fairness among processes.
Disadvantages: Increased overhead due to frequent context switching.

Priority Scheduling

Description: Processes are executed based on their priority (lower number = higher priority).
Advantages: Critical processes are executed first.
Disadvantages: May cause starvation of low-priority processes.

Implementation Details

Input:

1. Scheduling algorithm type.
2. For each process:
    - Arrival time
    - Burst time
    - Priority (if using Priority Scheduling(0 if null or not Priority Scheduling))
3. Time quantum (if using Round Robin).

Output:

- Overall Metrics:
    - Average waiting time.
    - Average turnaround time.
- Gantt Chart: Visual representation of process execution order.

Tools and Technologies Used

- **Programming Language:** Python
- **Libraries Used:** matplotlib (for Gantt chart visualization) & tkinter (for interactive GUI)
- **Development Environment:** VS Code

Flow of the Program

1. User selects a scheduling algorithm from the GUI.
2. Processes are added with parameters like arrival time, burst time, and priority.
3. Scheduler executes the chosen algorithm, calculates metrics, and generates a Gantt chart.
4. Results are displayed in the GUI, including the Gantt chart, average waiting time, and turnaround time.

Key Metrics

- **Average Waiting Time:** The mean of waiting times of all processes.
- **Turnaround Time:** The total time taken from arrival to completion for each process.

Interface

Interface Overview

Below is a image of the GUI, where users can input process data, select algorithms, and view results:

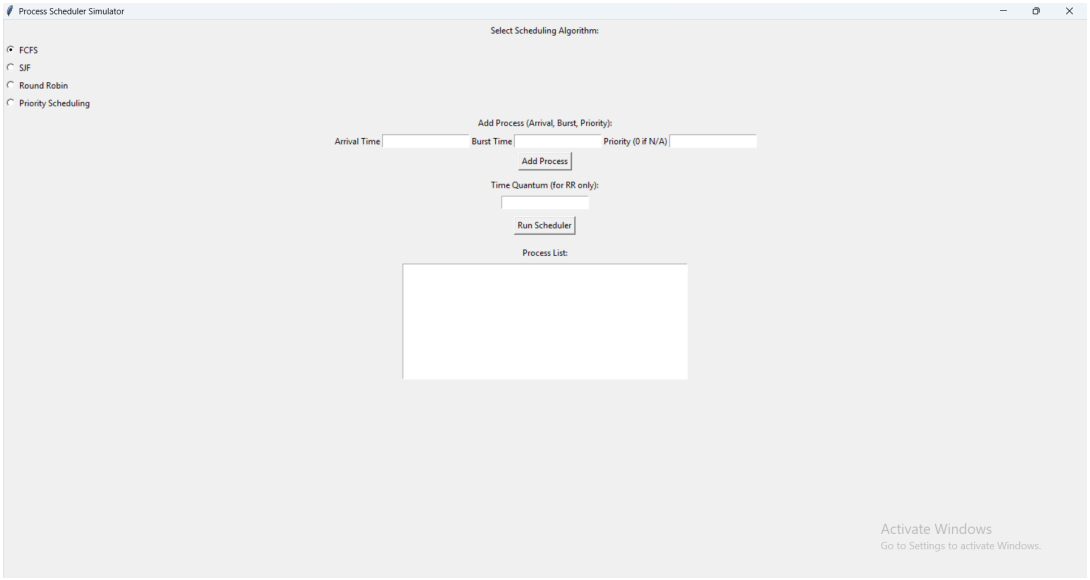

Figure 1: Graphical User Interface of the Process Scheduler Simulator

Simulation in Action

The following image shows the results of scheduling processes using the First Come First Serve algorithm:

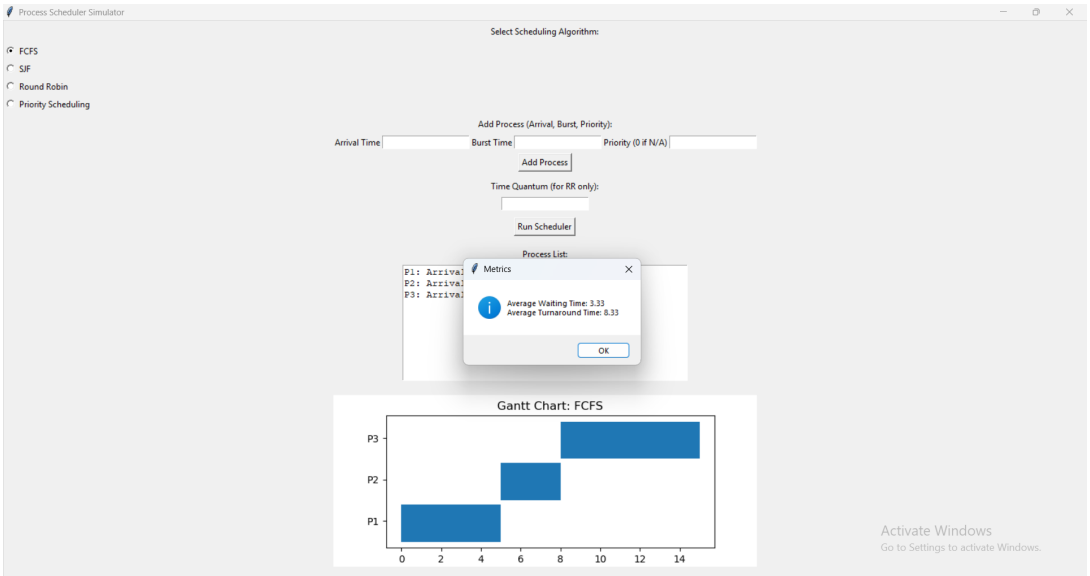

Figure 2: Scheduling Processes and Displaying Gantt Chart

Conclusion

This project successfully demonstrates the implementation of various scheduling algorithms and provides a practical tool for learning and experimentation. The GUI allows users to visualize and compare the performance of different scheduling approaches effectively.

Future Work

Future enhancements could include:

- Adding support for additional scheduling algorithms such as Multilevel Queue and Multilevel Feedback Queue.
- Integrating advanced visualization tools for better insights.
- Enhancing the GUI for mobile and web-based access.

References

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*.
2. Python Official Documentation.
3. Matplotlib Library Documentation.
4. Tkinter Library Documentation.