Operating Systems Project 1: Process Scheduling Simulation

Sunwoo Yeom, Arka Gantait, Rahul Kadam, Obaid Khan

Overview

This project focuses on implementing and comparing two CPU scheduling algorithms: First-In-First-Out (FIFO) and Shortest Job First (SJF). These algorithms would determine the execution order of processes in a system, affecting efficiency, turnaround time, and waiting time.

Implementation Details

- **First In, First Out (FIFO)**

    - Processes are scheduled based on arrival time.

    - Easy to execute, but can lead to higher waiting times for longer processes.

- **Shortest Job First (SJF)**

    - Requires sorting processes by burst time.

    - Results in lower average waiting and turnaround times when compared to FIFO.

    - Requires knowledge of burst times in advance, making it more practical in batch processing environments.

Results

To evaluate and compare FIFO and SJF, we tested them using a set of sample processes that had various arrival and burst times. The results emphasize the differences in average waiting time (AWT) and average turnaround time (ATT).

Test Case

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	3

FIFO Results

Process	Start Time	Completion Time	Turnaround Time	Waiting Time
P1	0	8	8	0
P2	8	12	11	7
P3	12	21	19	10
P4	21	24	21	18

- **Average Waiting Time (AWT): 8.75 ms**
- **Average Turnaround Time (ATT): 14.75 ms**

SJF Results

Process	Start Time	Completion Time	Turnaround Time	Waiting Time
P1	**0**	**8**	**8**	**0**
P2	**8**	**12**	**11**	**7**
P4	**12**	**17**	**14**	**9**
P3	**17**	**26**	**24**	**15**

- **Average Waiting Time (AWT): 7.75 ms**
- **Average Turnaround Time (ATT): 14.25 ms**

Comparison

- SJF performed better than FIFO by decreasing waiting and turnaround times.

- The improvement is due to prioritizing shorter tasks, which reduces overall processing delays.

- SJF requires knowing burst times in advance, which is not always practical in real-world situations.

Challenges & Solutions

1. Handling Process Arrival & Order: Verified accurate ordering through sorting for SJF.

2. Implementing SJF Correctly: Applied sorting algorithms to establish the order of execution.

3. Verifying Outputs: Compared results with expected scheduling behavior.