**Hassan Naeem**

**Operating systems**

**Report Project 1**

**4320-6320**

**Spring 2025**

**Operating Systems Project: Process Scheduling Simulation**

**1. Overview** This project implements two fundamental CPU scheduling algorithms:

* **First-Come, First-Served (FCFS)**: A non-preemptive scheduling algorithm where processes are executed in the order they arrive.
* **Shortest Job First (SJF)**: A non-preemptive scheduling algorithm where processes with the shortest burst time are executed first.

These algorithms are crucial for process management in operating systems, influencing system performance and response time.

**2. Implementation Details** The scheduling algorithms were implemented in C, following these key steps:

* **Process Structure**: Each process is defined by a Process structure containing the process ID (PID), arrival time, burst time, and priority.
* **File Input Handling**: Processes are read from an input file (processes.txt), which stores process details in tabular form.
* **FCFS Implementation**:
  + Sort processes by arrival time.
  + Compute waiting time (WT) and turnaround time (TAT).
  + Display the Gantt chart and performance metrics.
* **SJF Implementation**:
  + Sort processes by burst time.
  + Calculate waiting and turnaround times similarly to FCFS.
  + Handle tie-breaking using arrival time.

The implementation ensures correct execution order and efficient calculation of scheduling metrics.

**3. Results and Performance Comparison** Sample runs demonstrate the behavior of FCFS and SJF:

* **FCFS** leads to longer waiting times if shorter processes arrive later.
* **SJF** minimizes the waiting time and turnaround time by prioritizing short jobs.
* **SJF** performs better in scenarios with varying burst times.

**4. Challenges and Solutions** During the project, several challenges arose:

* **File Handling Errors**: Initially, the input file (processes.txt) was missing, causing the program to fail.

**Solution**: Verified file existence using ls -l and manually created it.

* **Compilation Issues**: The program threw errors due to loop variable declarations inside for loops in C90 mode.

**Solution**: Used the -std=c99 flag for compilation.

* **Incorrect Output**: Early versions of SJF did not handle tie-breaking correctly. **Solution**: Modified sorting logic to prioritize arrival time in case of equal burst times.

**Conclusion** This project successfully implemented FCFS and SJF scheduling algorithms. The analysis shows that SJF is generally more efficient than FCFS in reducing waiting time and turnaround time. Future improvements could include implementing **Round Robin** and **Priority Scheduling** for further comparisons.