King Saud University
College of Computer and Information Sciences
Information Technology Department

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Programming Assignment 1

Assignment Report

Section#:52586 Group#: 4	
Name	**ID**
Ghaida Alhussain (Leader)	**444201213**
Fajer Alamro	**444200800**
Sara Alhowaimel	**444200462**
Deem Aljarba	**444200523**

Supervised by: Dr. Amani AL-Ahmadi

King Saud University
College of Computer and Information Sciences
Information Technology Department

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Task distribution:

Name	Task
Ghaida Alhussain	Implemented the Process class and debugging, contributed to report writing (Implementation section).
Fajer Alamro	Implemented part of the Scheduling class, contributed to report writing (Demonstration of FCFS and preemption in SRTF scheduling section)
Sara Alhowaimel	Implemented part of the Scheduling class, contributed to report writing (Introduction and Conclusion sections).
Deem Aljarba	Implemented OSProject &Display Classes, contributed to report writing (Execution Process and Performance Analysis section)

King Saud University
College of Computer and Information Sciences
Information Technology Department

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Table of Contents

King Saud University
College of Computer and Information Sciences
Information Technology Department

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Introduction

Efficient process scheduling is a fundamental aspect of modern operating systems, ensuring optimal CPU utilization and minimizing process waiting time. This project focuses on implementing the Shortest Remaining Time First (**SRTF**) scheduling algorithm, incorporating First-Come, First-Served (**FCFS**) scheduling for processes with equal CPU burst times.

The goal is to analyze the performance of this scheduling approach by evaluating key metrics such as CPU utilization, average turnaround time, and average waiting time. Additionally, the project illustrates how context switching affects scheduling efficiency and demonstrates the effectiveness of combining SRTF and FCFS to handle processes with varying execution requirements.

King Saud University
College of Computer and Information Sciences
Information Technology Department

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Implementation

The program consists of four main classes:

- **OSProject Class:** Handles user input and initializes processes.
- **Process Class:** Represents a Process with the following attributes:
    - **ID**: A unique identifier for each process.
    - **ArrivalTime**
    - **BurstTime**
    - **remainingTime**
    - **completionTime**
    - **waitingTime:** calculated as:

      Waiting Time=Turnaround Time−Burst Time
    - **turnaroundTime:** calculated as:

      Turnaround Time=Completion Time−Arrival Time

- **Scheduling Class:** Implements the SRTF scheduling algorithm with FCFS for processes with equal remaining time, executes processes, and calculates performance metrics. It's also included attributes such that:
    - **processes:** A list that holds all processes to be scheduled.
    - **ganttChart:** A list that stores the execution order of processes for visualization.
    - **totalWaitingTime**: Sum of all waiting times for performance evaluation.
    - **totalTurnaroundTime**: Sum of all turnaround times for performance evaluation.
    - **totalIdleTime**: The total CPU idle time when no process is executing.

- **Display Class**: Displays results, including:
    - Process details.
    - The scheduling order using a Gantt Chart.
    - Performance metrics such as Average Turnaround Time, Average Waiting Time, and CPU Utilization.

King Saud University
College of Computer and Information Sciences
Information Technology Department

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CSC227: Operating System

Execution Process and Performance Analysis

1. The program prompts the user to enter the number of processes, followed by the **arrival time** and **burst time** for each process. These inputs are then stored in a list for scheduling.

```
Enter the number of processes: 2
Enter arrival time and burst time for P1:
2 4
Enter arrival time and burst time for P2:
3 1
```

Figure 1:Sample Run

Process 1 (P1): Arrival Time = 2 ms, Burst Time = 4 ms

Process 2 (P2): Arrival Time = 3 ms, Burst Time = 1 ms

The program then applies Shortest Remaining Time First **(SRTF)** scheduling, using First-Come, First-Served **(FCFS)**when two processes have the same remaining time.

2. After scheduling the processes, the program generates the following **Gantt Chart**, which represents the execution timeline:

```
Time    Process/CS
0–2     IDLE
2–3     P1
3–4     CS
4–5     P2
5–6     CS
6–9     P1
```

Figure 2: Sample Run

Time	Process/CS/Idle
0-2 ms	The CPU remains **IDLE** because no process has arrived yet.
2-3 ms	**P1 starts execution**, but since P2 arrives at 3ms and has a shorter burst time, the CPU preempts P1 and switches to P2.
3-4 ms	A **context switch** (CS) occurs to transfer control to P2.
4-5 ms	**P2 executes** and completes, since it only required 1ms**.**
5-6 ms	Another **context switch** occurs before P1 resumes execution.
6-9 ms	**P1 resumes execution** and completes at 9ms**.**

This demonstrates how SRTF prioritizes the process with the shortest remaining time, leading to preemption when a shorter job arrives.

3. After execution, the program calculates key performance metrics:

```
Performance Metrics
Average Turnaround Time: 4.50
Average Waiting Time: 2.00
CPU Utilization: 33.33%
BUILD SUCCESSFUL (total time: 7 seconds)
```

Figure 3: Sample Run

- **Average Turnaround Time (TAT):** On average, each process spends 4.5ms from arrival to completion. This is relatively low due to SRTF's ability to minimize wait times.

- **Average Waiting Time (WT):** The average waiting time is 2.0ms, meaning that processes experience moderate delays before execution starts.

- **CPU Utilization:** The CPU was actively executing processes 33.33% of the time.

CSC227: Operating System

Demonstration of FCFS and Preemption in SRTF Scheduling

```
run:
Enter the number of processes: 4
Enter arrival time and burst time for P1:
0 8
Enter arrival time and burst time for P2:
1 4
Enter arrival time and burst time for P3:
2 2
Enter arrival time and burst time for P4:
3 7
----------------------------------------
P1 : Arrival time = 0 , Burst time = 8  ms
P2 : Arrival time = 1 , Burst time = 4  ms
P3 : Arrival time = 2 , Burst time = 2  ms
P4 : Arrival time = 3 , Burst time = 7  ms

Scheduling Algorithm: Shortest Remaining Time First
Context Switch: 1 ms

Time    Process/CS
0-1     P1
1-2     CS
2-4     P3
4-5     CS
5-9     P2
9-10    CS
10-17   P1
17-18   CS
18-25   P4

Performance Metrics
Average Turnaround Time: 12.25
Average Waiting Time: 7.00
CPU Utilization: 84.00%
BUILD SUCCESSFUL (total time: 14 seconds)
```

Figure 4: Demonstration of Preemption in SRTF

- **At time 0:** P1 arrives and starts execution since it is the only available process.
- **At time 1:** P2 arrives (burst time = 4 ms), and since it has a **shorter burst time** than P1 (4 ms vs. 7 ms remaining**), P1 is preempted**, and a context switch occurs.
- **At time 2:** P3 arrives (burst time = 2 ms), and since it has a **shorter burst time** than P2 (2 ms vs. 3 ms remaining), **P2 is preempted**, and a context switch occurs.
- **At time 4:** P3 completes execution, and a context switch occurs.
- **At time 5:** P2 resumes execution and continues running, then CS occurs at time 9.
- **At time 10:** P1 resumes execution and continues running, then CS occurs at time 17.
- **At time 18:** P4 starts execution.
- **At time 25:** P4 completes execution, and all processes are finished.

```
run:
Enter the number of processes: 3
Enter arrival time and burst time for P1:
2 3
Enter arrival time and burst time for P2:
4 3
Enter arrival time and burst time for P3:
6 2
----------------------------------------
P1 : Arrival time = 2 , Burst time = 3  ms
P2 : Arrival time = 4 , Burst time = 3  ms
P3 : Arrival time = 6 , Burst time = 2  ms

Scheduling Algorithm: Shortest Remaining Time First
Context Switch: 1 ms

Time    Process/CS
0-2    IDLE
2-5    P1
5-6    CS
6-8    P3
8-9    CS
9-12   P2

Performance Metrics
Average Turnaround Time: 4.33
Average Waiting Time: 1.67
CPU Utilization: 50.00%
BUILD SUCCESSFUL (total time: 14 seconds)
```

Figure 5: Demonstration of FCFS in SRTF for Equal Burst Times

- P1 arrives first (2 ms) and starts execution.
- At 4 ms, P2 arrives, but it has the same burst time as P1 (3 ms).
- Since P1 arrived first, it continues execution (FCFS rule).
- At 6 ms, P3 arrives with a shorter burst time (2 ms), so P1 is preempted.
- P3 executes and completes.
- P2 resumes execution after P3 completes, following the FCFS order.

Conclusion

Finally, we realized that while the SRTF scheduling algorithm effectively reduces waiting and turnaround times by prioritizing shorter processes, it also increases context switching overhead, which negatively impacts CPU efficiency. Reducing the number of preemptions could improve CPU utilization, especially as the number of processes grows. Additionally, incorporating FCFS for processes with equal burst times ensures fair execution, prevents starvation, and maintains process order. This highlights the importance of balancing efficiency, fairness, and CPU performance in process scheduling to achieve optimal results.