King Saud University

College of Computer and Information Sciences
Department of Information Technology

Programming Assignment1

Process Scheduling Simulator

CSC 227 Semester-2, 1446H

Student names and IDs:

Student's Names	Student's IDs	Section
Maha Albakr	444201108	78287
Hatoun Ibrahim Almogherah	444203015	78287
Rama Khalid Alomair	444200662	78287
Walaa Saif Aleslam	444200088	78287

Supervised By: Dr. Rabia Jafri

Task Distribution:

Student's Names	Task	Description
Hatoun	Process and Data Structures, Event-Driven Simulation	Defining the **Process class** and managing **process-related data structures**. implement the **event-driven simulation**, ensuring that process execution follows a time-based approach where processes arrive, execute, and complete based on the scheduling algorithm.
Maha Albakr	Performance Metrics Calculation & Output Visualization	Calculate key **performance metrics**, including **CPU utilization, average turnaround time, and average waiting time**. handle the **final output display**, ensuring that performance results are formatted and presented clearly.
Rama	Scheduling Algorithm (Preemptive SJF with FCFS tie-breaker)	implements the **Shortest Remaining Time First (SRTF)** scheduling algorithm with **FCFS as a tie-breaker** when two processes have the same remaining burst time. ensure that the CPU always selects the process with the shortest remaining execution time while correctly handling preemptions.
Walaa Saif Aleslam	Context Switching & Gantt Chart	manages **context switching**, ensuring that a **1 ms delay** is introduced whenever a process switch occurs. Build and display the **Gantt Chart**, visually representing process execution over time, including context switches and preemptions.

Screen shots showing sample input/output:

i. The FCFS condition is demonstrated (i.e., some processes have equal next CPU bursts)

Input: (P4 has same burst time as P3)

```
Number of processes= 4
Arrival times and burst times as follows:
P1 Arrival time = 0
P1 Burst time = 8
P2 Arrival time = 1
P2 Burst time = 4
P3 Arrival time = 2
P3 Burst time = 5
P4 Arrival time = 3
P4 Burst time = 5
```

Output: (P3 runs first because it arrived before P4)

```
Scheduling Algorithm: Shortest remaining time first
Context Switch: 1 ms

Time  Process/CS
0-1 P1
1-2 CS
2-6 P2
6-7 CS
7-12 P3
12-13 CS
13-18 P4
18-19 CS
19-26 P1

Performance Metrics
Average Turnaround Time: 14.00
Average Waiting Time: 8.50
CPU Utilization: 84.62%
```

ii. Preemption is demonstrated (a newly arriving process has a shorter next CPU burst than the CPU burst time remaining for the currently executing process)

Input: (P2 arrives after P1 but has shorter burst)

```
Number of processes= 3
Arrival times and burst times as follows:
P1 Arrival time = 0
P1 Burst time = 8
P2 Arrival time = 1
P2 Burst time = 3
P3 Arrival time = 2
P3 Burst time = 6
```

Output: (P2 preempts P1 because it has a shorter burst time)

```
Scheduling Algorithm: Shortest remaining time first
Context Switch: 1 ms

Time  Process/CS
0-1 P1
1-2 CS
2-5 P2
5-6 CS
6-12 P3
12-13 CS
13-20 P1

Performance Metrics
Average Turnaround Time: 11.33
Average Waiting Time: 5.67
CPU Utilization: 85.00%
```