**National University of Computer & Emerging Sciences, Karachi Computer Science Department**



**Spring 2025**

**Lab 7 Tasks**

|  |  |
| --- | --- |
| **Course Code: CL-2005** | **Course: Operating Systems Lab (CL 2006)** |
| **Instructor(s):** | **Fatima Gado** |

## Task 1: First-Come, First-Served (FCFS) Scheduling

Solve using the FCFS scheduling algorithm. Calculate the average waiting time (AWT) and average turnaround time (ATAT).

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 0 | 6 |
| P2 | 2 | 8 |
| P3 | 4 | 7 |
| P4 | 6 | 3 |

## Task 2: Shortest Job First (SJF) – Non-Preemptive

Solve using the SJF (Non-Preemptive) scheduling algorithm. Calculate the average waiting time (AWT) and average turnaround time (ATAT).

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 1 | 5 |
| P2 | 3 | 2 |
| P3 | 5 | 9 |
| P4 | 6 | 4 |

## Task 3: Shortest Remaining Time First (SRTF) – Preemptive SJF

Solve using the SRTF (Preemptive SJF) scheduling algorithm. Calculate the average waiting time (AWT) and average turnaround time (ATAT).

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 0 | 9 |
| P2 | 1 | 4 |
| P3 | 2 | 7 |
| P4 | 3 | 2 |

## Task 4: Round Robin (RR) Scheduling – Time Quantum = 3ms

Solve using the Round Robin scheduling algorithm with time quantum = 3ms. Calculate the average waiting time (AWT) and average turnaround time (ATAT).

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 0 | 10 |
| P2 | 1 | 5 |
| P3 | 2 | 8 |
| P4 | 3 | 3 |

## Task 5: Round Robin (RR) Scheduling – Time Quantum = 2ms

Solve using the Round Robin scheduling algorithm with time quantum = 2ms. Calculate the average waiting time (AWT) and average turnaround time (ATAT).

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 0 | 12 |
| P2 | 2 | 6 |
| P3 | 4 | 8 |
| P4 | 6 | 4 |

## Task 6: Priority Scheduling (Non-Preemptive)

Solve using the Priority Scheduling (Non-Preemptive) algorithm (Lower number = higher priority).

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival Time | Burst Time | Priority/Queue |
| P1 | 0 | 7 | 3 |
| P2 | 1 | 4 | 1 |
| P3 | 2 | 6 | 2 |
| P4 | 3 | 5 | 4 |

## Task 7: Priority Scheduling (Preemptive)

Solve using the Priority Scheduling (Preemptive) algorithm (Lower number = higher priority).

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival Time | Burst Time | Priority/Queue |
| P1 | 0 | 8 | 2 |
| P2 | 1 | 3 | 1 |
| P3 | 2 | 5 | 3 |