Multilevel Queue Scheduling

Nihara .W.A.S.

22001344

Introduction:

This report discusses the implementation and results of a program that Multilevel Queue scheduling algorithms. The implemented algorithms include Round **Robin, Shortest Job First, and First-In-First-Out (FIFO)**. The program allows users to input the details of multiple processes, including burst time and priority, and then simulates the execution of these processes using the specified scheduling algorithms.

Implementation:

The program is implemented in C programming language. It consists of several functions to perform different tasks such as enqueue and dequeue processes, implementing scheduling algorithms, and handling user inputs. The main functionalities of the program are as follows:

Enqueue and dequeue processes into different priority queues based on their priority levels.

Round Robin (priority level 0): This algorithm iterates through the highest priority queue (q0) and allocates a fixed time slice (quantum time) to each process. If the process completes within the time slice, it exits the queue. Otherwise, it rejoins the back of the queue with its remaining time.

Shortest Job First (priority levels 1 and 2): This algorithm prioritizes processes with the shortest remaining burst time. It identifies the process with the shortest remaining time from all non-empty queues and executes it until completion or until the available time slice is exhausted.

First In First Out (priority level 3): This algorithm processes jobs in the order they arrive. It processes the process at the front of the highest priority queue (q0) until completion or until the available time slice is exhausted

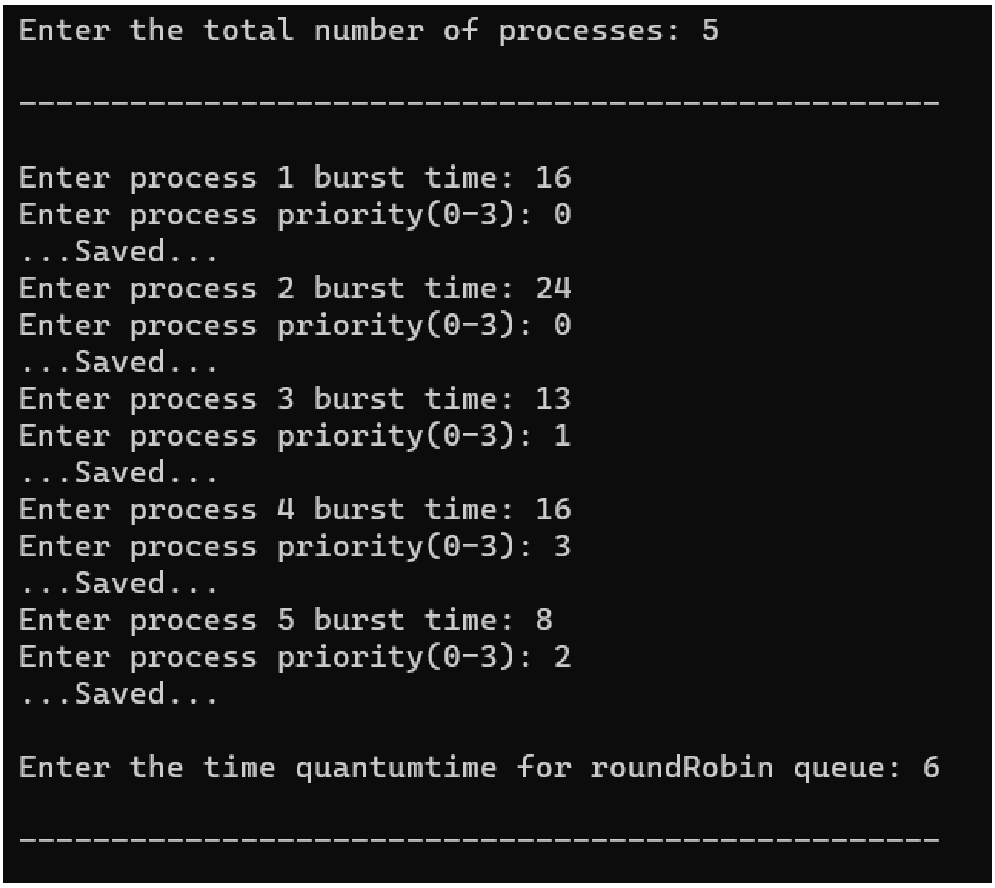
The program iterates through these algorithms in a specific order (q0 -> q1 -> q2 -> q3) until all processes are completed.

Prompt users to input the total number of processes, burst time, and priority for each process and Allow users to specify the time quantum for the Round Robin scheduling algorithm.

Results:

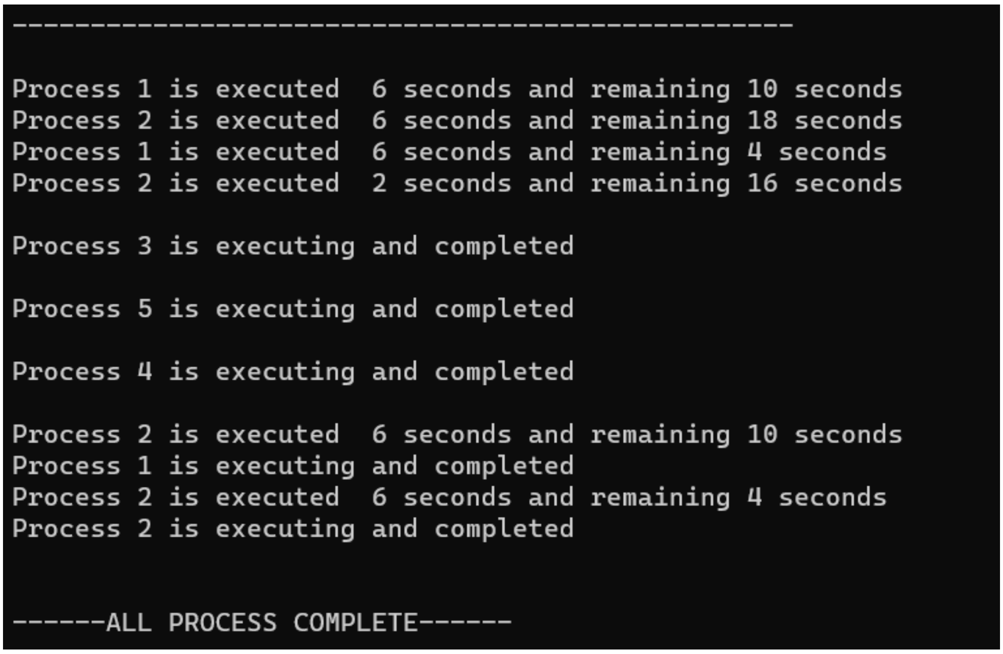
The output of the program includes detailed information about each process, such as its ID, execution time, and remaining time. Additionally, the program prints messages indicating when a process starts executing and when it completes.

Sample Inputs:



```
Enter the total number of processes: 5

--------------------------------------------------


Enter process 1 burst time: 16
Enter process priority(0-3): 0
...Saved...
Enter process 2 burst time: 24
Enter process priority(0-3): 0
...Saved...
Enter process 3 burst time: 13
Enter process priority(0-3): 1
...Saved...
Enter process 4 burst time: 16
Enter process priority(0-3): 3
...Saved...
Enter process 5 burst time: 8
Enter process priority(0-3): 2
...Saved...

Enter the time quantumtime for roundRobin queue: 6

--------------------------------------------------
```

Sample Outputs:



```
--------------------------------------------------

Process 1 is executed  6 seconds and remaining 10 seconds
Process 2 is executed  6 seconds and remaining 18 seconds
Process 1 is executed  6 seconds and remaining 4 seconds
Process 2 is executed  2 seconds and remaining 16 seconds

Process 3 is executing and completed

Process 5 is executing and completed

Process 4 is executing and completed

Process 2 is executed  6 seconds and remaining 10 seconds
Process 1 is executing and completed
Process 2 is executed  6 seconds and remaining 4 seconds
Process 2 is executing and completed


------ALL PROCESS COMPLETE------
```

Pros and Cons of Each Scheduling Algorithm:

- Round Robin:

Pros: Ensures fairness in CPU allocation by allowing each process to execute for a fixed quantum of time.

Cons: May lead to starvation for longer processes if the quantum time is too small.

- Shortest Job First:

Pros: Minimizes average waiting time and turnaround time by prioritizing processes with shorter burst times.

Cons: May lead to starvation for longer processes if short processes keep arriving.

- First-In-First-Out (FIFO):

Pros: Simple and easy to implement.

Cons: Can lead to starvation for shorter processes if longer processes arrive first.

Conclusion:

The implemented program provides a simulation of different process scheduling algorithms, allowing users to observe their behavior and performance. Each scheduling algorithm has its own advantages and disadvantages, and the choice of algorithm depends on the specific requirements and constraints of the system. While Round Robin ensures fairness, Shortest Job First minimizes waiting time, and FIFO is simple but may lead to longer waiting times for some processes.

Limitations:

The program assumes a fixed time quantum for Round Robin scheduling, which may not be optimal for all scenarios.

It does not consider factors such as I/O operations, process arrival times, or preemption.