Report

OS Simulation based Assignment

Course Code: CSE316

Submitted by:

Aman Kumar

12106661

RK21GPA06

Question no. 6

Write a program for multilevel queue scheduling algorithm. There must be three queues generated. There must be specific range of priority associated with every queue. Now prompt the user to enter number of processes along with their priority and burst time. Each process must occupy the respective queue with specific priority range according to its priority. Apply Round Robin algorithm with quantum time 4 on queue with highest priority range. Apply priority scheduling algorithm on the queue with medium range of priority and First come first serve algorithm on the queue with lowest range of priority. Each and every queue should get a quantum time of 10 seconds. CPU will keep on shifting between queues after every 10 seconds.

Introduction

- Multilevel queue scheduling is a type of CPU scheduling in which the processes in the ready state are divided into different groups, each group having its own scheduling needs.

- The ready queue is divided into different queues according to different properties of the process like memory size, process priority, or process type. All the different processes can be implemented in different ways, i.e., each process queue can have a different scheduling algorithm

- Each queue has its own priority level and scheduling algorithm to manage the processes inside that queue.

- Queues can be arranged in a hierarchical structure.

- High-priority queues might use Round Robin scheduling.

- Low-priority queues might use First Come, First Serve scheduling.

- This algorithm prioritizes different types of processes and ensures fair resource allocation.

Algorithm/Logic to solve the problem

- To solve the given problem first we will initiate three queues and associate specific range of priority with every queue. First queue will have more priority than the rest two queues. In this program, Higher the priority number, Higher the priority.

- Enter number of processes along with their priority and burst time. Here arrival time is assumed to be 0 for all processes. The process whose details will be entered first will be assumed to be enter inside the respective queue first.

- Apply Round Robin Algorithm with quantum time 4 unit on highest priority queue.

- Apply Priority Scheduling Algorithm on medium priority queue.

- Apply First Come First Serve on lowest priority queue.

- Each queue will get only 10 unit time because we have to apply round robin algorithm on the whole queue hierarchical structure with quantum time 10 unit.

For example

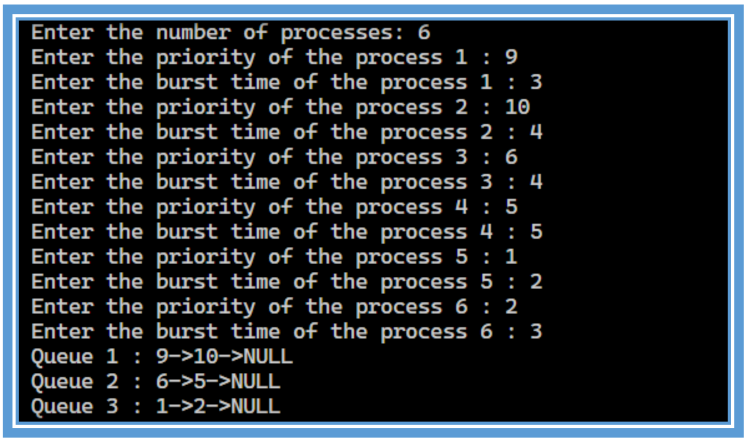
q1: p1, p2, p3 |RR (4) |

q2: p4, p5, p6 |PS ------ | Round Robin (10)

q3: p7, p8, p9 |FCFS |

Screenshots

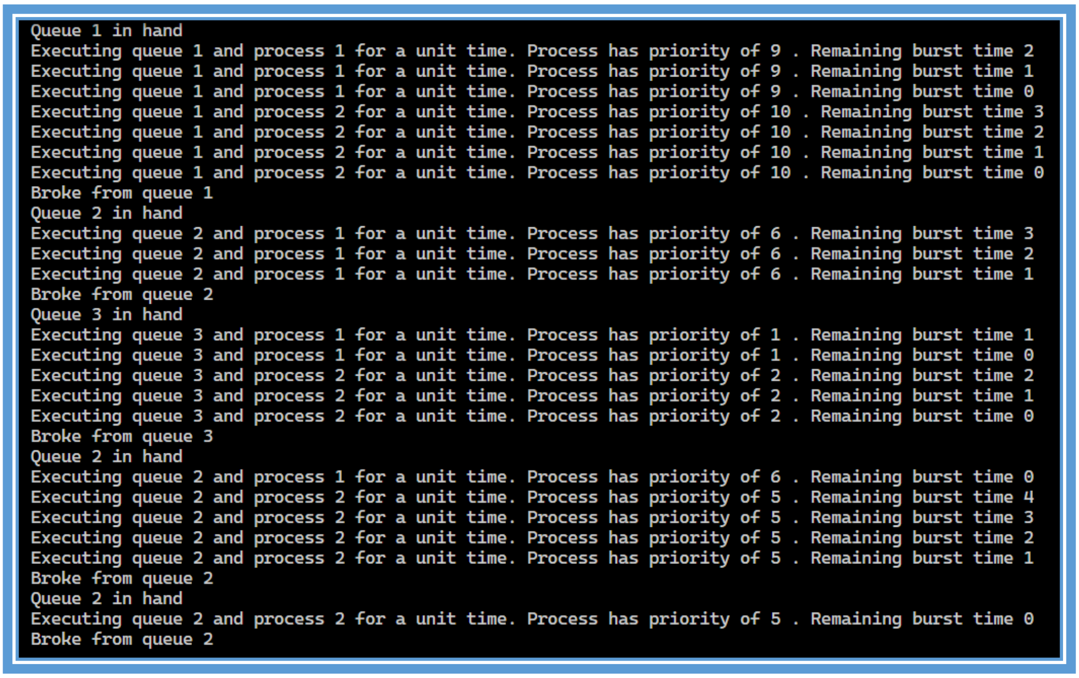
This section of output is taking user input and after that displaying the structure of each queue.



```
Enter the number of processes: 6
Enter the priority of the process 1 : 9
Enter the burst time of the process 1 : 3
Enter the priority of the process 2 : 10
Enter the burst time of the process 2 : 4
Enter the priority of the process 3 : 6
Enter the burst time of the process 3 : 4
Enter the priority of the process 4 : 5
Enter the burst time of the process 4 : 5
Enter the priority of the process 5 : 1
Enter the burst time of the process 5 : 2
Enter the priority of the process 6 : 2
Enter the burst time of the process 6 : 3
Queue 1 : 9->10->NULL
Queue 2 : 6->5->NULL
Queue 3 : 1->2->NULL
```

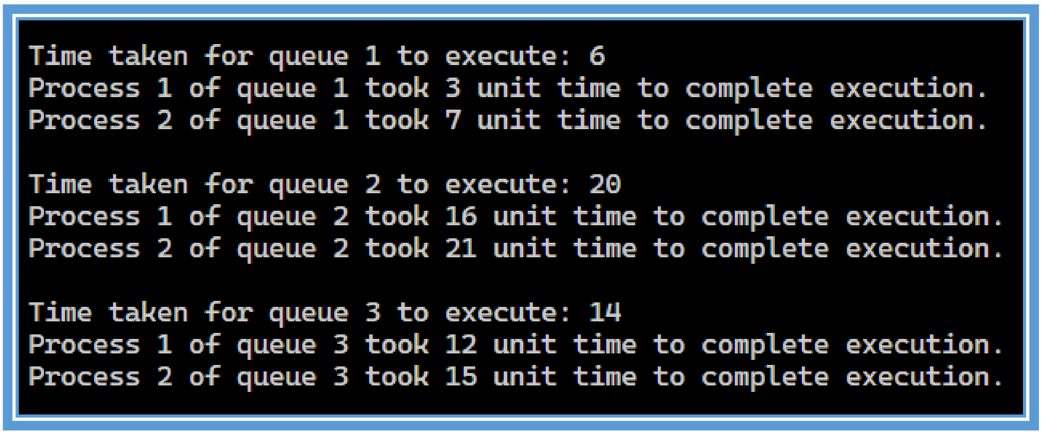
.

This section is showing the execution of each queue for one unit time and breaks from the queue if either all processes are completed of that particular queue or the quantum time expires.



```
Queue 1 in hand
Executing queue 1 and process 1 for a unit time. Process has priority of 9 . Remaining burst time 2
Executing queue 1 and process 1 for a unit time. Process has priority of 9 . Remaining burst time 1
Executing queue 1 and process 1 for a unit time. Process has priority of 9 . Remaining burst time 0
Executing queue 1 and process 2 for a unit time. Process has priority of 10 . Remaining burst time 3
Executing queue 1 and process 2 for a unit time. Process has priority of 10 . Remaining burst time 2
Executing queue 1 and process 2 for a unit time. Process has priority of 10 . Remaining burst time 1
Executing queue 1 and process 2 for a unit time. Process has priority of 10 . Remaining burst time 0
Broke from queue 1
Queue 2 in hand
Executing queue 2 and process 1 for a unit time. Process has priority of 6 . Remaining burst time 3
Executing queue 2 and process 1 for a unit time. Process has priority of 6 . Remaining burst time 2
Executing queue 2 and process 1 for a unit time. Process has priority of 6 . Remaining burst time 1
Broke from queue 2
Queue 3 in hand
Executing queue 3 and process 1 for a unit time. Process has priority of 1 . Remaining burst time 1
Executing queue 3 and process 1 for a unit time. Process has priority of 1 . Remaining burst time 0
Executing queue 3 and process 2 for a unit time. Process has priority of 2 . Remaining burst time 2
Executing queue 3 and process 2 for a unit time. Process has priority of 2 . Remaining burst time 1
Executing queue 3 and process 2 for a unit time. Process has priority of 2 . Remaining burst time 0
Broke from queue 3
Queue 2 in hand
Executing queue 2 and process 1 for a unit time. Process has priority of 6 . Remaining burst time 0
Executing queue 2 and process 2 for a unit time. Process has priority of 5 . Remaining burst time 4
Executing queue 2 and process 2 for a unit time. Process has priority of 5 . Remaining burst time 3
Executing queue 2 and process 2 for a unit time. Process has priority of 5 . Remaining burst time 2
Executing queue 2 and process 2 for a unit time. Process has priority of 5 . Remaining burst time 1
Broke from queue 2
Queue 2 in hand
Executing queue 2 and process 2 for a unit time. Process has priority of 5 . Remaining burst time 0
Broke from queue 2
```

This section is showing the overall stats of each queue after execution

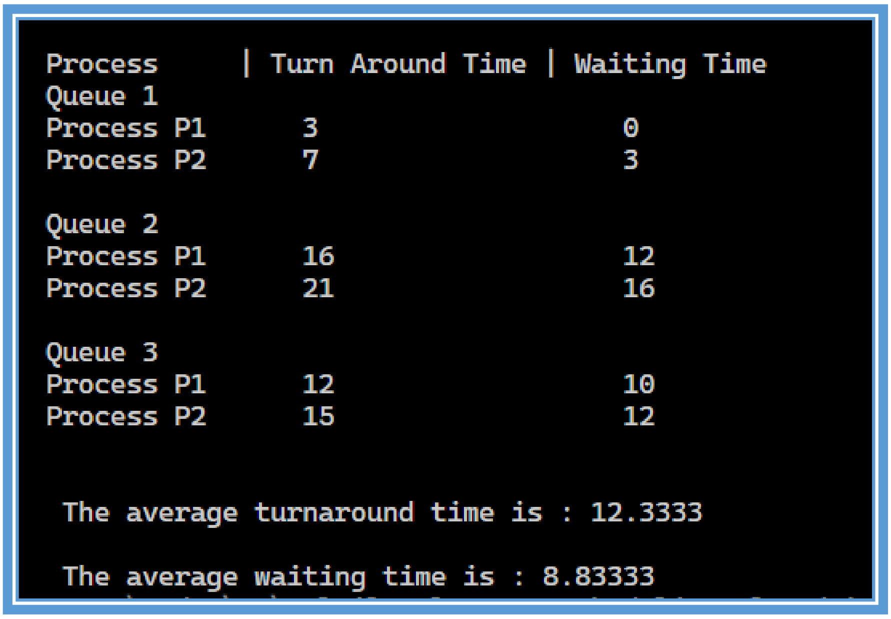


```
Time taken for queue 1 to execute: 6
Process 1 of queue 1 took 3 unit time to complete execution.
Process 2 of queue 1 took 7 unit time to complete execution.

Time taken for queue 2 to execute: 20
Process 1 of queue 2 took 16 unit time to complete execution.
Process 2 of queue 2 took 21 unit time to complete execution.

Time taken for queue 3 to execute: 14
Process 1 of queue 3 took 12 unit time to complete execution.
Process 2 of queue 3 took 15 unit time to complete execution.
```

This section is showing the turn around time and waiting time for each process and at the end the average of them.



```
Process      | Turn Around Time | Waiting Time
Queue 1
Process P1      3                   0
Process P2      7                   3

Queue 2
Process P1      16                  12
Process P2      21                  16

Queue 3
Process P1      12                  10
Process P2      15                  12


 The average turnaround time is : 12.3333

 The average waiting time is : 8.83333
```

GitHub Link

multilevel-queue-scheduling-algorithm

Thank You