Multilevel Queue

Abeywickrama I T

22000072

Introduction

The implemented program simulates a multilevel queue scheduling algorithm with 4 queue, each assigned a priority from q0 to q3.it use Round Robin(RR) scheduling for q0, shortest job first (SJF) for q1,q2, and first in first out(FIFO) for q3. The user enters the number of processes, their priorities, and burst time. The program prints the order of execution and waiting time and turnaround time for each process.

Waiting time= amount of time that process spend waiting in the ready queue before executed.

Turnaround time= time taken from the submission of a process to its completion, including waiting time and execution time.

Implementation

First it execute q0 which has higher priority for 20s . It has round robin scheduling. Each process get a small unit of CPU time. After this time has elapsed, the process is preempted and added to the end of the ready queue.

After 20s it will switch to q1. It has shortest job first scheduling. CPU in assigned to processes with smallest next CPU burst.

After 20s, it will switch to q3. It also has shortest job first scheduling.

After 20s. it will switch to q4. It has First-In-First-out scheduling. The process that came first will execute first.

After 20s, it will switch to q0 again and soon.

Pros and Cons of Each Scheduling Algorithm

1.Round Robin (RR):

Pros: Ensure fairness among processes by providing quantem time.

Cons: May have higher waiting times for processes with longer burst time

2.Shortest Job First (SJF):

Pros: associates with each process the length of its next CPU burst. CPU is assigned to process with smallest next CPU burst.

Gives minimum average waiting time

Cons: may lead to starvation for longer processes if shorter ones continuously.

Not suitable for time-sensitive applications

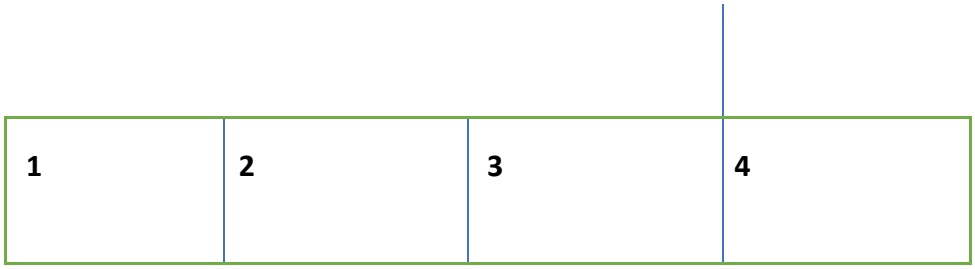
3.First-In-First-Out (FIFO):

Pros: Simple to implement. Guarantees fairness in term of execution order.

Cons: May result in longer waiting time and turnaround time for longer processes. Not suitable for time sensitive applications.

Explain with example

If we entered 11 processes like below, q0-(RR)
quantem time=4



Burst time=6 3 8 5

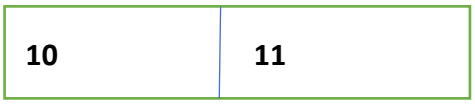
q1-(SJF)

burst time= 10 5 15

q2-(SJF)

burst time= 3 5

q3-(SJF)



burst time= 10 15

switch time=20s

It will switch between queues after every 20s. q0 having a highest priority and q3 having lowest priority.

In round robin algorithm we take 4s as quentem time. After this time has elapsed , the process is preempted and added to the end of the ready queue.

Entered processes like,

```
Enter the total number of processes:11
Enter the burst time of process 1 :6
Enter the priority of the process 1 (0 or 1 or 2 or 3):0
Enter the burst time of process 2 :3
Enter the priority of the process 2 (0 or 1 or 2 or 3):0
Enter the burst time of process 3 :8
Enter the priority of the process 3 (0 or 1 or 2 or 3):0
Enter the burst time of process 4 :5
Enter the priority of the process 4 (0 or 1 or 2 or 3):0
Enter the burst time of process 5 :10
Enter the priority of the process 5 (0 or 1 or 2 or 3):1
Enter the burst time of process 6 :5
Enter the priority of the process 6 (0 or 1 or 2 or 3):1
Enter the burst time of process 7 :15
Enter the priority of the process 7 (0 or 1 or 2 or 3):1
Enter the burst time of process 8 :3
Enter the priority of the process 8 (0 or 1 or 2 or 3):2
Enter the burst time of process 9 :5
Enter the priority of the process 9 (0 or 1 or 2 or 3):2
Enter the burst time of process 10 :10
Enter the priority of the process 10 (0 or 1 or 2 or 3):3
Enter the burst time of process 11 :15
Enter the priority of the process 11 (0 or 1 or 2 or 3):3
Enter the quantem time for round robin algorithm:4
```

It executed like below,

```
Process 2 is finished successfully
Process 1 is finished successfully
Process 3 is running for 3 second
Process 6 is finished successfully
Process 5 is finished successfully
Process 8 is finished successfully
Process 9 is finished successfully
Process 10 is finished successfully
Process 3 is finished successfully
Process 4 is finished successfully
Process 7 is finished successfully
Process 11 is finished successfully
```

Each processes' waitingTime and Turnaround time shows like below,

```
Process WaitingTime     Turnaround Time
1       11              17
2       4               7
3       61              69
4       65              70
5       25              35
6       20              25
7       65              80
8       40              43
9       43              48
10      48              58
11      70              85
```