Multilevel Queue Scheduling Algorithm

Another class of scheduling algorithms has been created for situations in which processes are easily classified into different groups.

For example, A common division is made between foreground(or interactive) processes and background (or batch) processes. These two types of processes have different response-time requirements, and so might have different scheduling needs. In addition, foreground processes may have priority over background processes.

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.

For example, separate queues might be used for foreground and background processes. The foreground queue might be scheduled by the Round Robin algorithm, while the background queue is scheduled by an FCFS algorithm.

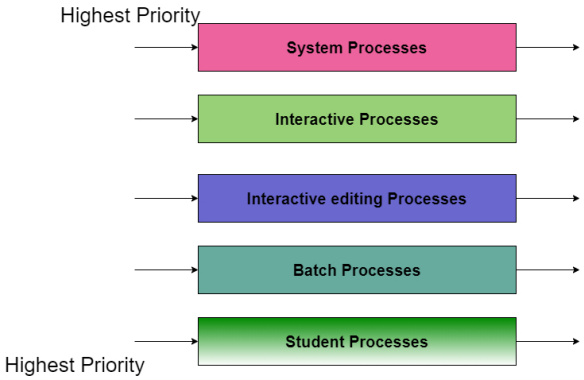
In addition, there must be scheduling among the queues, which is commonly implemented as fixed-priority preemptive scheduling. **For example,** The foreground queue may have absolute priority over the background queue.

Let us consider an example of a multilevel queue-scheduling algorithm with five queues:

1. System Processes
2. Interactive Processes
3. Interactive Editing Processes
4. Batch Processes

5. Student Processes

Each queue has absolute priority over lower-priority queues. No process in the batch queue, for example, could run unless the queues for system processes, interactive processes, and interactive editing processes were all empty. If an interactive editing process entered the ready queue while a batch process was running, the batch process will be preempted.



In this case, if there are no processes on the higher priority queue only then the processes on the low priority queues will run. For **Example:** Once processes on the system queue, the Interactive queue, and Interactive editing queue become empty, only then the processes on the batch queue will run.

The Description of the processes in the above diagram is as follows:

- **System Process** The Operating system itself has its own process to run and is termed as System Process.
- **Interactive Process** The Interactive Process is a process in which there should be the same kind of interaction (basically an online game).
- **Batch Processes** Batch processing is basically a technique in the Operating system that collects the programs and data together in the form of the **batch** before the **processing** starts.
- **Student Process** The system process always gets the highest priority while the student processes always get the lowest priority.

In an operating system, there are many processes, in order to obtain the result we cannot put all processes in a queue; thus this process is solved by Multilevel queue scheduling.

Implementation

Given below is the C implementation of Multilevel Queue Scheduling:


```c
#include<stdio.h>
int main()
{
    int p[20],bt[20],su[20],wt[20],tat[20],i,k,n,temp;
```

```
    float wtavg, tatavg;
    printf("Enter the number of processes:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        p[i] = i;
        printf("Enter the Burst Time of Process%d:", i);
        scanf("%d",&bt[i]);
        printf("System/User Process (0/1) ? ");
        scanf("%d", &su[i]);
    }
    for(i=0;i<n;i++)
        for(k=i+1;k<n;k++)
            if(su[i] > su[k])
            {
```

Output

The output of the above code is as follows:

```
Enter the number of processes:3
Enter the Burst Time of Process0:12
System/User Process (0/1) ? 0
Enter the Burst Time of Process1:18
System/User Process (0/1) ? 0
Enter the Burst Time of Process2:15
System/User Process (0/1) ? 1

PROCESS          SYSTEM/USER PROCESS    BURST TIME      WAITING TIME    TURNAROUND TIME
0                0                      12              0               12
1                0                      18              12              30
2                1                      15              30              45
Average Waiting Time is --- 14.000000
Average Turnaround Time is --- 29.000000
```

Advantages of Multilevel Queue Scheduling

With the help of this scheduling we can apply various kind of scheduling for different kind of processes:

For System Processes: First Come First Serve(FCFS) Scheduling.

For Interactive Processes: Shortest Job First (SJF) Scheduling.

For Batch Processes: Round Robin(RR) Scheduling

For Student Processes: Priority Scheduling

Disadvantages of Multilevel Queue Scheduling

The main disadvantage of Multilevel Queue Scheduling is the problem of starvation for lower-level processes.

Let us understand **what is starvation?**

Starvation:

Due to starvation lower-level processes either never execute or have to wait for a long amount of time because of lower priority or higher priority process taking a large amount of time.

← Prev

Next →

Game Development
Data Structure & Alog.
Operating System
Computer Network
Computer Architecture
More...