**MULTI LEVEL QUEUE SCHEDULING**

Pre-initialized conditions:

* Time quantum is directly initialized to 4 in the code. It can be changed if any other time quantum requirement popup.
* All the arrays like completion time, waiting time, burst time are initialized with 500.
* Arrival time is taken as 0 for all the process.
* Priority contains 0 and 1.1 Indicates high priority and 0 indicates low priority.

Time taken for code execution: 313 milliseconds= 0.313 seconds

Method used for time calculation:

*long startTime = System.currentTimeMillis();*

*program....*

*long endTime = System.currentTimeMillis();*

*long totalTime = endTime - startTime;*

*System.out.println(totalTime);*

Proof of correctness of algorithm:

Algorithm:

* Burst time and priority of each process is read from the excel sheet and saved into two arrays of size 500.
* Based on the priority the all the process is moved into 2 queues.
* The process with high priority (1) is moved to queue 1 and all the process with low priority are moved to queue 2.
* First come First serve (FCFS) algorithm is performed on queue 1. If all the process in queue are completed, then the execution will move to queue 2.
* Round Robin scheduling algorithm is used on queue 2.
* After the execution of both the queues is completed then we calculate waiting time and turnaround time of each process.

Input:

* Process [0,1,2,3,4]
* Burst time [3,4,7,2,4]
* Priority [0,1,1,0,1]
* Arrival time =0
* Time quantum = 3

Example:

* Q1-----Process [1,2,4]
* Q2-----Process [0,3]
* FCFS on Q1
* RR on Q2

Final Gant chart is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P0 | P3 |
| 0 4 | 4 11 | 11 15 | 15 18 | 18 20 |

Average Waiting time: 9.6

Average Turnaround time: 13.6