A Study on Performance Analysis of Multi-Level Feedback Queue Scheduling Approach

Sanjeeva Polepaka, R. P. Ram Kumar

Abstract*: In CPU scheduling, various algorithms are used to schedule the processes. Few of them are First come first serve (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF), Priority Scheduling, Round Robin (RR), Multi-Level Queue (MLQ), Multi-Level Feedback Queue (MLFQ) scheduling approaches. This scheduling is used to process the scheduling of operating systems, which is responsible for assigning the CPU time to available processes. To get user interactivity, throughput, real-time responsiveness, and more. The objective of the paper is to present an idea that keeps the CPU in maximum utilization until the process is requesting for an event. When the process is waiting for an event to occur, the CPU is switched between the processes for better utilization by consuming CPU cycles. The paper also addresses the four different approaches and their average waiting time in processing the jobs.

Index Choice: CPU Scheduling, Process Scheduling, First come first Serve (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF), Round Robin(RR), Multilevel Feedback Queue (MLFQ), Waiting Time.

I. INTRODUCTION

Multilevel Feedback Queuing is common in the CPU scheduling techniques used in operating systems. Multi-Level Feedback Queue (MLFQ) algorithm allows processes switching between queues based on the burst time. The MLFQ scheduling algorithm allows a process to move between queues. Here, the processes are categorized based on CPU burst time. If too much of CPU cycles are utilized by the process, then it is given less priority and moved to the lower-priority queue. This scheme leaves I/O-bound and interactive processes in the higher-priority queues. Also, a process that waits too long in a lower-priority queue may be moved to a higher-priority queue. This form of aging prevents starvation. This paper suggested an improvised MLFQ with decreased waiting time.

Let us consider an example of multilevel feedback queue scheduler which contains three queues with numbering 0 to 2 as shown in the Figure 1. The processes will be first executed in queue 0. Next, the processes will be completed in queue one if queue 0 is empty. Similarly, processes will be executed in queue two if queue 0 and queue 1 are empty.

Let us assign time quantum of 0.8 milliseconds to queue 0 and time quantum 0.16 millisecond to queue 1. The processes will be run in queue two on FCFS basic but the condition is queue 0 and one should be empty.

Revised Manuscript Received on December 28, 2018.

Sanjeeva Polepaka, Associate Professor, [2]Department of Computer Science and Engineering, Malla Reddy Engineering College (Autonomous), Maisammaguda, Secunderabad – 500100, India.

R. P. Ram Kumar, Professor, [2]Department of Computer Science and Engineering, Malla Reddy Engineering College (Autonomous), Maisammaguda, Secunderabad – 500100, India.

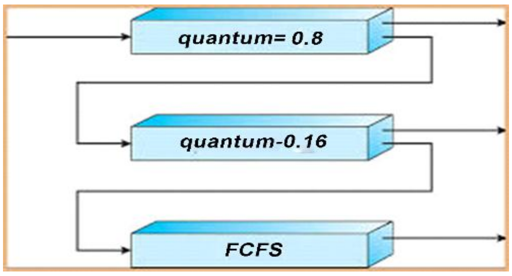

Fig 1: FCFS Scheduling Method

II. THE PROPOSED WORK

Figure 2 shows the proposed method with the queue 0 and 1 submitted at the same time quantum. But in queue 2 the processes will run on basis of Shortest Remaining Time First (SRTF) order which will reduce the waiting time of the processes. By this, the processes will be executed quickly without any delay.

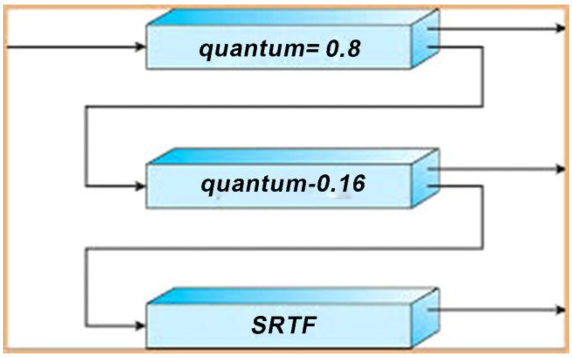

Fig 2: SRTF Scheduling Method

Example Case 1:

Consider the following example. Let us take four processes P_1, P_2, P_3, and P_4 with arrival time 0, 3, 10, 12 simultaneously and estimate time of 10, 5, 3, and 1, simultaneously.

Equation (1) is used to find the waiting time by the FCFS approach.

WT = ST – AT (1)

where WT represents Waiting time,
ST represents Starting time, and
AT represents Arrival time.

	Arrival time	**Estimate time**	**Starting time**	**Waiting time**
P1	0	10	0	0
P2	3	5	10	7
P3	10	3	15	5
P4	12	1	18	6

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In FCFS approach, the determined average waiting time is 4.5 milli seconds (ms)

Example Case 2:

Equation (2) is used to find the waiting time by SJF approach.

WT = ST – AT (2)

	Arrival time	Estimate time	Starting time	Waiting time
P1	0	10	0	0
P2	3	5	10	7
P3	10	3	15	5
P4	12	1	18	6

In SJF approach, the determined average waiting time is again 4.5 ms

Example Case 3:

Equation (3) is used to find the waiting time by SRTF approach.

WT = TAT – ET (3)

where WT represents Waiting time

TAT represents Turn around time

ET represents Estimate time.

TAT is calculated using the equation (4).

TAT = CT – AT (4)

where, CT represents Completion Time and

AT represents Arrival Time

	AT	ET	ST	CT	WT	TAT
P1	0	10	0, 8, 14	19	9	19
P2	3	5	3	8	0	5
P3	10	3	10	13	0	3
P4	12	1	13	14	1	2

In SRTF approach, determined Average waiting time is 2.5 ms

Example Case 4:

Equation (4) is used to find the waiting time by Round Robin Fashion:

WT = TAT – ET and TAT = CT – AT (5)

	AT	ET	ST	CT	WT	TAT
P1	0	10	0,2,6,10,16	18	8	18
P2	3	5	4,8,14	15	7	12
P3	10	3	12,18	19	6	9
P4	12	1	15	16	3	4

Average waiting time = 6 ms

III. CONCLUSION

Experimental results concluded that, when determining the average waiting time for four approaches, namely, FCFS, SJF, SRTF, and RR approaches, SRTF has the minimum average waiting time in processing the jobs. By this, it is clear that by using Shortest Remaining Time First the waiting time of the processes is reduced and the processes will be executed quickly without any delay.

REFERENCES

1. Malhar Thombare, Rajiv Sukhwani, Priyam Shah"Efficient implementation of Multilevel Feedback Queue Scheduling”, International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), September 2016.
2. Silberchatz, Galvin, and Gagne, operating systems concepts, 8th Edition, John Wiley and Sons, 2009.
3. Deepali Maste, Leena Ragha, Nilesh Marathe, "Intelligent Dynamic Time Quantum Allocation IMLFQ Scheduling" in International Journal of Information and Computation Technology, vol. 3, no. 4, pp. 311-322, 2013, International Research Publications House, ISBN 0974-2239.
4. A. S. Tanenbaum, “Modern Operating Systems” Prentice Hall Publications, 2009.