The Performance Analysis of CPU scheduling Algorithms in Operating Systems

Thangakumar Jeyaprakash[1,*] , Ranjana P[2] and Sambath M[3]
[1]*Hindustan Institute of Technology and Science, Chennai, Padur, India*
[2]*Hindustan Institute of Technology and Science, Chennai, Padur, India*
[3]*Hindustan Institute of Technology and Science, Chennai, Padur, India*
[*]Corresponding: *tkumar@hindustanuniv.ac.in*

Abstract
Scheduling algorithms plays a significant role in optimizing the CPU in operating system. Each scheduling algorithms schedules the processes in the ready queue with its own algorithm design and its properties. In this paper, the performance analysis of First come First serve scheduling, Non preemptive scheduling, Preemptive scheduling, Shortest Job scheduling and Round Robin algorithm has been discussed with an example and the results has been analyzed with the performance parameters such as minimum waiting time, minimum turnaround time and Response time.

Keywords:

CPU scheduling, Waiting time, Turnaround time, Response Time.

1. Introduction

Operating system [3] schedules the pool of processes[7] to execute in processor from ready to run state based upon the arrival time, priority and the time quantum. Few scheduling algorithms such as First come first Serve (FCFS) scheduling, Preemptive priority scheduling, Non-Preemptive priority scheduling, Shortest Job First (SJF) Scheduling and Round Robin [2] scheduling (RR) which executes the processes using various parameters like arrival time, priority and time quantum. Each process must transit different states during execution such as new, ready, run and termination. The execution of a process may not be completed for two reasons. First one, if a process requires an I/O request [1] and the other one, the time quantum expires. This quantum property makes a way to other processes in the queue with a distributed chance. Figure 1 represents the classical machine with time quantum to speed up the CPU tasks effectively

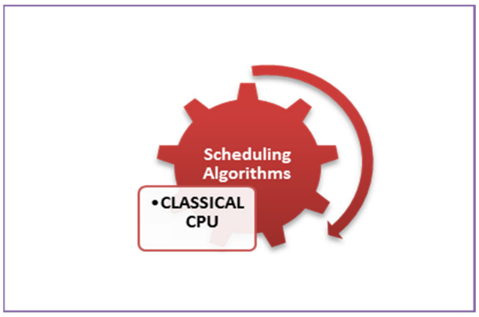

Figure 1: Classical CPU with Scheduling Algorithms

Whenever the time slice expires, the executing process will be moved out and the next process will be dispatched by the short-term scheduler to the processor for execution. This quantum process will speed up the response time of each process for a successful completion.

Materials and methods

First Come First Serve Scheduling

In this scheduling, the process come first in the ready queue will be moved to execution which is also known as Batch processing. The response time of each process depends on the arrival time and completion time of the process arrived earlier. In FCFS scheduling, the processes are non – preemptive. In non-preemption, the process cannot be forcibly stopped, and it can be stopped only voluntarily once its completed the task.

In the below example,

Process no (P.N0) – Process ID., ms - milliseconds

Arrival time (AT) – The time of the process arrived at the ready queue

Burst Time (BT) – The time required for the process to execute

Completion time (CT) – The time required for the process to complete

Turnaround time (TAT) – The difference between the completion time and arrival time

Manuscript received July 5, 2023
Manuscript revised July 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.7.19

Waiting time (WT) – The difference between the Turnaround time and burst time

P NO	AT(ms)	BT(ms)	CT(ms)	TAT(ms)	WT(ms)	RT(ms)
P1	0	4	4	4	0	0
P2	1	3	7	6	3	3
P3	2	1	8	6	5	5
P4	3	2	10	7	5	5
P5	4	5	15	11	6	6

Table 1: First come first serve Scheduling

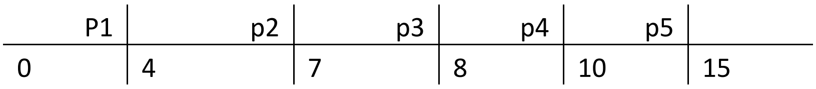

Figure2: Gantt chart of FCFS

From the Gantt chart, it is observed that the response time of each process in first come first serve scheduling is little higher. The average turnaround time and the waiting time are 6.8 and 3.8 respectively.
The average response time is 5.8 ms.

Non - Preemptive Priority scheduling

In preemptive priority scheduling, a process cannot be forcibly swapped with the new process. The process will be executed in the CPU based on the priority

In the below example,

Process no (P.N0) – Process ID., ms - milliseconds

Arrival time (AT) – The time of the process arrived at the ready queue

Burst Time (BT) – The time required for the process to execute

Completion time (CT) – The time required for the process to complete

Turnaround time (TAT) – The difference between the completion time and arrival time

Waiting time (WT) – The difference between the Turnaround time and burst time

P NO	AT(ms)	Priority	BT(ms)	CT(ms)	TAT(ms)	WT(ms)	RT(ms)
P1	0	(l)2	4	4	4	0	0
P2	1	4	3	15	14	11	11
P3	2	6	1	12	10	9	9
P4	3	10	2	6	3	1	1
P5	4	8(h)	5	11	7	2	2

Table 2: Preemptive Scheduling

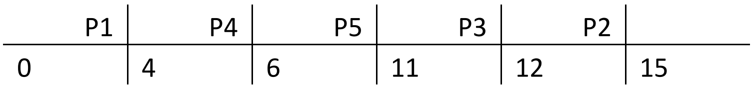

Figure3: Gantt chart of Non-Preemptive Scheduling

The average turnaround time and the waiting time are 7.6ms and 4. 6 ms respectively. The average response time of each process is 4.6 ms which is much higher than preemptive priority scheduling

Preemptive Priority scheduling

In preemptive priority scheduling, a process can be forcibly swapped with the new process based upon the priority even though the process has not completed its execution.

In the below example,

Process no (P.N0) – Process ID., ms - milliseconds

Arrival time (AT) – The time of the process arrived at the ready queue

Burst Time (BT) – The time required for the process to execute

Completion time (CT) – The time required for the process to complete

Turnaround time (TAT) – The difference between the completion time and arrival time

Waiting time (WT) – The difference between the Turnaround time and burst time

P NO	AT(ms)	Priority	BT(ms)	CT(ms)	TAT(ms)	WT(ms)	RT(ms)
P1	0	(l)2	4	15	15	11	0
P2	1	4	3	12	11	8	0
P3	2	6	1	3	1	0	0
P4	3	10	2	5	2	0	0
P5	4	8(h)	5	10	6	1	1

Table 4: Non - Preemptive Scheduling

P1	P2	P3	P4	P5	P2	P1	
0	1	2	3	5	10	12	15

Figure5: Gantt chart of FCFS

From the Gantt chart, it is observed that the response time of each process in Preemptive priority scheduling is little lesser than the First come first serve scheduling and Non preemptive scheduling. The average turnaround time and the waiting time are 7 and 4respectively.The average response time of each process is 0.2 ms only

Shortest Job First (SJF Scheduling)

In Shortest job first[4], a process with the shortest execution time will be executed first.

In the below example,

Process no (P.N0) – Process ID., ms - milliseconds

Arrival time (AT) – The time of the process arrived at the ready queue

Burst Time (BT) – The time required for the process to execute

Completion time (CT) – The time required for the process to complete

Turnaround time (TAT) – The difference between the completion time and arrival time

Waiting time (WT) – The difference between the Turnaround time and burst time

P NO	AT(ms)	BT(ms)	CT(ms)	TAT(ms)	WT(ms)	RT(ms)
P1	0	4	4	4	0	0
P2	1	3	15	14	11	6
P3	2	1	12	10	9	2

P4	3	2	6	3	1	2
P5	4	5	11	7	2	6

Table 4: SJF Scheduling

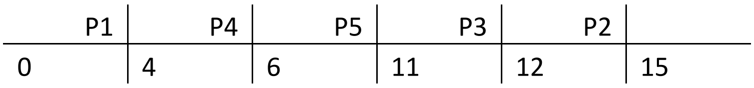

Figure 6: Gantt chart of SJF Scheduling

From the Gantt chart, it is observed that the response time of each process in SJF scheduling is effective than the other scheduling algorithms. The average turnaround time and the waiting time are 6.2 ms and 3.2 ms respectively. The average response time of each process is 3.2 ms respectively.

Round Robin Scheduling (RR Scheduling)

In Round Robin scheduling[9][10], each process is preemptive which refers to multitasking[5]. If a process has been forcibly stopped from execution and dispatch the other process due to time quantum expires, then the concept is known as Multi-tasking.
In the below example,

P No – Process number,

Arrival time (AT) – The time of the process arrived at the ready queue

Burst Time (BT) – The time required for the process to execute

Completion time (CT) – The time required for the process to complete

Turnaround time (TAT) – The difference between the completion time and arrival time

Waiting time (WT) – The difference between the Turnaround time and burst time

In this algorithm, each process will be executed till a time quantum expires and shift the next queuing process to the execution. The process which was forcibly stopped will be moved to the queue for the next chance, this process will continue, till the completion of each process.

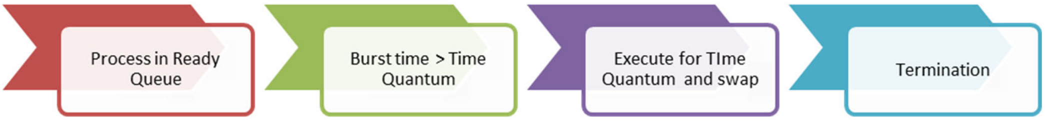

Figure 7: Scheduling with Time Quantum

In the following example with time quantum = 2 units, we have shown how the process executed with time quantum[6] of 2 units. If the time quantum is less than two units, it will be completely executed or else it swaps the next process and moves the preemptive process in queue for the next cycle.

Initially, P1 arrives and the CPU schedules P1 for 2 units. Process P2 and P3 arrive during the execution of process P1, till the time quantum completes. Now the process P1 will be stopped from execution and moved to the ready queue and process P2 will be dispatched to the processor.

Time Quantum = 2						
P NO	AT(ms)	BT(ms)	CT(ms)	TAT(ms)	WT(ms)	RT(ms)
P1	0	4	7	7	3	0
P2	1	3	12	11	8	1
P3	2	1	5	3	2	2
P4	3	2	9	6	4	4
P5	4	5	15	11	6	5

Table 5: Round Robin Algorithm

The Gantt chart for the above example as follows

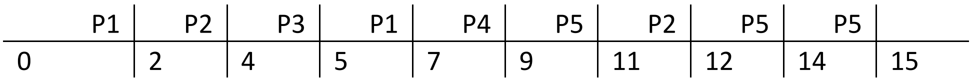

Figure 8: Gantt Chart of RR algorithm

This time quantum continues till the execution of all process in the pool of ready queue. The average Turnaround time of each process is 7.6 ms and the average waiting time of each process is 4.6 ms.

Results and discussion

In figure 9, the performance analysis of First come First serve scheduling, non-preemptive scheduling, Preemptive scheduling, Shortest Job scheduling and Round Robin algorithm has been analyzed from an example the performance parameters such as minimum waiting time, minimum turnaround time and Response time. It is observed that, Premptive priority algorithm has the minimum response time for the processes as the process with the highest priority will be responded immediately. The SJF scheduling algorithm has the minimum waiting time as the process with the shortest job will be executed first in CPU. The SJF has the minimum turnaround time compared to all the algorithm as shown in the figure 9.

Scheduling Algorithms	Avg. TAT (in ms)	Avg. WT (in ms)	Avg. RT (in ms)
FCFS	6.8	3.8	3.8
NP - Priority	7	4	0.2
P - Priority	7.6	4.6	4.6
SJF	6.2	3.2	3.2
RR	7.6	4.6	2.4

Table 6: Performance measures

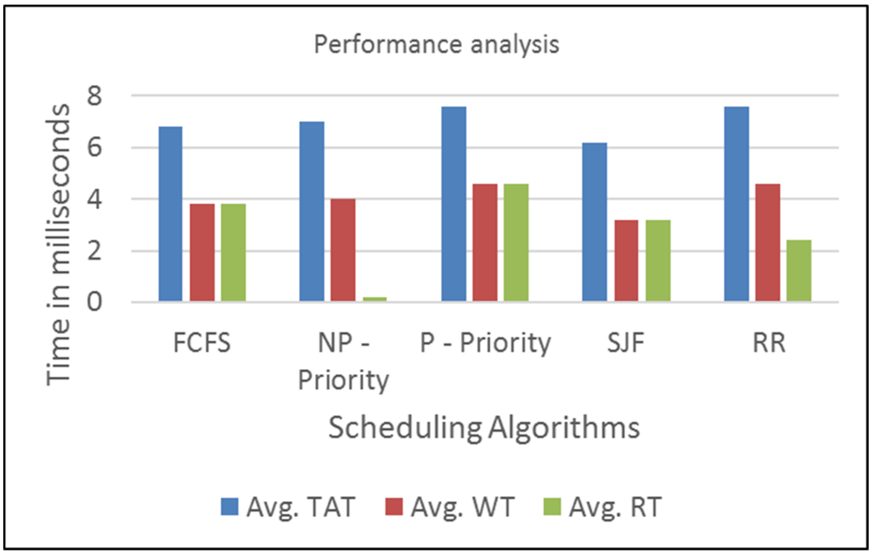

Figure 9. Performance measures of Scheduling Algorithms

Conclusions

In this paper, the performance analysis of First come First serve scheduling, Non preemptive scheduling, Preemptive scheduling, Shortest Job scheduling and Round Robin algorithm has been discussed with an example and the results has been analyzed with the performance parameters such as minimum waiting time, minimum turnaround time and Response time.

References

[1] Matthias Möller & Cornelis Vuik, On the impact of quantum computing technology on future developments in high-performance scientific computing, DOI - 10.1007/s10676-017-9438-0Ethics and Information Technology

[2] Abbas Noon et al, A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average, International Journal of Computer Science Issues, Vol 8.

[3] Henry Corrigan et al, Quantum Operating Systems, HotOS '17: Proceedings of the 16th Workshop on Hot Topics in Operating Systems May 2017 Pages 76–81https://doi.org/10.1145/3102980.310299

[4] Neetu Goel, Dr. R. B. Garg, “A Comparative Study of CPU Scheduling Algorithms”, International Journal of Graphics & Image Processing, Vol 2, issue 4, November 2012

[5] Alban Allkoci, Elona Dhima, Igli Tafa, “Comparing Priority and Round Robin Scheduling Algorithms”, IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 3, No 1, May 2014.

[6] Ashok Kumar, Dr. Harish Rohil, Suraj Arya, “Analysis of CPU Scheduling Policies through Simulation”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, ISSN: 2277 128X, May 2013.

[7] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, John Wiley & Sons, United States, 2005.

[8] Andrew S. Tanenbaum, Albert S. Woodhull, “Operating Systems Design and Implementation”, Prentice Hall of India, New Delhi, 2010

[9] Haidar M. Ali, Kaies Khalid, “An Improvement on Round Robin Scheduling Method” International Conference on Information Technology and Natural Sciences, ICITNS-2003

[10] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash and Himanshu Sharma, “An Improved Round Robin Schedduling Algorithm for CPU Scheduling”, International Journal on Computer Science and Engineering, Vol. 02, No.04, 2010, pp. 1064-1066.