Divide & Quantum

Muhammad Hassaan Rafiq¹, Ameer Hamza²

^{1,2}Lahore Garrison University, Lahore, Pakistan

¹hassaan.rafiq@lgu.edu.pk, ²ameerhamza844@gmail.com

Abstract:

When the operating system was introduced to the world, many functions of it were also introduced that included process management, file management, memory management, networking. As the operating system tends to interact with different operations at the runtime so keeping that in mind we can say that the process management is most essential in an operating system because it allows the operating system to interact with different operations more efficiently and it also improves the timing of the operating systems to interact between the processes. In order to manage between processes many algorithms were introduced by different people and different minds and today we find this as an opportunity develop something new by keeping the previous algorithms in mind and compare them and find a best possible result.

1. INTRODUCTION

CPU is a primary computer resource. So, its scheduling is central to operating system design. To improve both utilization of CPU several process would be kept in the memory unit at a time meaning that multi-programming concept is used here. Operating system should allow several processes to run at a single time so that the CPU utilization could be maximized. With the help of multiprogramming utilize this time to maximize the utilization of the CPU as few processes wait in the memory unit while one process is being executed.

According to Safaei, S.Jha, D.Platt, almost all the computer are scheduled before use. For better utilization and better performance of system CPU scheduling is important [1].

Previously people have done so much research in the CPU scheduling field like how to get the maximum performance from the CPU, how to utilize CPU with all means, how to deal with number of maximum process as in possible maximum short time of period.

2. RELATED WORK

There are many people who already

contributed in the improvement of the CPU scheduling algorithms. Like an Improved Round Robin Algorithm in which an improved version of round robin, the time quantum is doubled on each turn [2]. Another Optimized Round Robin in which the time quantum is doubles second time the queue is traversed from the beginning and then the 3rd time the shortest job is processed first [3]. And then the 1st and 2nd cycle is considered again. Similarly, another Dynamic Quantum with Re adjusted Round Robin Scheduling algorithm proposed by Behra and its team in which they used the median of the burst times of the processes and used it as quantum time [4].

Abdulrazaq Abdulrahim and his team introduced another algorithm a NIRRR [5]. He introduced 2 new queues in it and introduced dynamic quantum by taking the average of the burst times of the processes in the queues. Another Algorithm for real time systems priority-based Algorithm the quantum is letting k and the processes are processed according to the priority and then the processes are given new priorities according to their remaining time [6].

3. SCHEDULING CRITERIA

The criteria for the CPU scheduling

include the following:

1. CPU Utilization. We want to keep the CPU as busy as possible.
 2. Throughput. The number of processes completed per time unit is called throughput.
 3. Turnaround time. Turnaround time is the total time spent by a process till its execution.
 4. Waiting time. Waiting time is the time a process spend waiting in the ready queue.
 5. Response time. It is the time taken for the memory to respond to the call of a process.
- So it can be said that a good algorithm must include the following:

1. Maximize CPU utilization.
2. Maximize throughput.
3. Maximize turnaround time.
4. Minimize waiting time.
5. Minimize response time.

4. PROPOSED ALGORITHM

Keeping the scheduling criteria in mind our group have proposed a new algorithm that works on the divide and quantum name, it includes the following:

- I. The burst time of every process will be divided to maximize the interaction of the CPU among processes.
- II. If the burst will still be too much large so that it takes most of the time of CPU than quantum will be applied that will be half from the dividing condition.
- III. If the burst time will be already lesser than the dividing condition than it will be executed in one time without any dividing condition or any quantum being applied.
- IV. Till the completion of all the processes, the proposed algorithm will be repeated.

5. EXPERIMENTS

Table : Experiments Figures

Process	Burst Time
A	53
B	17
C	68
D	24

The quantum is 10 and the condition is 20 the condition is always two times than the quantum.

If the burst time of a process is greater than 20 then the burst time is divided by 2, if it still greater than 20 than quantum is applied but if it becomes less than 20 after being divided than the divided burst time will be executed completely.

But if the burst time is less than 20 than the process will be executed completely at once.

Gantt chart

Using Round Robin Technique.

Quantum = 10

A	B	C	D	A	B	C	D	A	C	D
10	20	30	40	50	57	67	77	87	97	101

A	C	A	C	A	C
111	121	131	141	144	162

Waiting time:

A = 144 - 53 = 91

B = 57 - 17 = 40

C = 162 - 68 = 94

D = 101 - 24 = 77

Total waiting time = 302

Avg. waiting time = 75.5

Using Proposed Algorithm.

A	B	C	D	A	C	D	A	C	A	C
10	27	37	47	63	73	87	100	114	128	162

Waiting time:

A = 128 - 53 = 75

B = 27 - 17 = 10

C = 162 - 68 = 94

D = 87 - 24 = 63

Total waiting time = 242

Avg. waiting time = 60.5

2.

Process	Arrival Time	Burst Time
A	0.0	3
B	1.0	6
C	4.0	4
D	6.0	2

The quantum is 2 and the condition is 4 because we have set the condition always two times than the quantum.

If the burst time of a process is greater than 4 then the burst time is divided by 2, if it is still greater than 4 than the quantum is applied but if it becomes less than 4 after being divided than the divided burst time will be executed completely.

But if the burst time is less than 4 than the process will be executed completely at once.

Gantt chart

Using Round Robin Technique.

Quantum = 2

Waiting time:

$$A = 9 - 3 = 6$$

$$B = 15 - 6 = 9$$

$$C = 13 - 4 = 9$$

$$D = 8 - 2 = 6$$

Total waiting time = 30

Avg. waiting time = 7.5

Using Proposed Algorithm.

Waiting time:

$$A = 3 - 3 = 0$$

$$B = 15 - 6 = 9$$

$$C = 10 - 4 = 6$$

$$D = 12 - 2 = 10$$

Total waiting time = 25

Avg. waiting time = 6.25

6. CONCLUSION

There is still a lot of research to be done

in the field of CPU scheduling because there are still some lacks that are present in previous algorithm. The 'DIVIDE & QUANTUM' results better than the round robin algorithm which is best among the previous algorithms. The proposed algorithm has beaten round robin in the research period done to check whether our algorithm is better or not. So, as it stands based on the results that have been found we can say that the proposed algorithm is better one.

7. REFERENCES

- [1] F. Safaei, S.Jha and D. Platt, 2005. Processing resources scheduling in programmable networks.
- [2] Manish Mishra, Abdul Kadir Khan, "An Improved Round Robin CPU Scheduling Algorithm", Journal of Global Research in Computer Science, Vol. 03, No.06, June 2012, ISSN-2229-371X.
- [3] Ajit Singh, Priyanka Goyal, Sahil Batra, "An optimized Round Robin Scheduling Algorithm for CPU Scheduling", International Journal on Computer Science and Engineering, Vol. 02, No. 07, 2010, 2383-2385.
- [4] Behra, H.S, Mohanty, R and Dabeshree, N (2010) "A new Proposed Dynamic Quantum with Re Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis", International Journal of Computer Applications (0975-8887) Volume 5, No. 5, pp 10-5.
- [5] Abdulrazaq Abdulrahim, Saleh E Abdullahi, Junaidu B. Sahalu, "A New Improved Round Robin CPU Scheduling Algorithm", International Journal of Computer Application (0975-8887), Vol. 90, No. 4, March 2014.
- [6] Ishwari Singh Rajput, Deepa Gupta, "A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", International Journal of Innovations in Engineering and Technology (IJET), Vol. 1, Issue 3, Oct 2012, ISSN: 2319-1058.