Dynamical Modified R.R. CPU Scheduling Algorithm

Amit Kumar Sain
Computer science Department
Rajasthan Technical University (Kota).

***Abstract*- Central Processing Unit (CPU) plays a significant role in computer system by transferring its control among different processes. As CPU is a central component, hence it must be used efficiently. Operating system performs an essential task, which is known as CPU scheduling used for efficient utilization of CPU.CPU Scheduling is a discipline which helps us gain deep insight into the complex set of policies and mechanisms used to govern the order in which tasks are executed by the processor. This paper proposes a Dynamic time quantum Shortest Job Round Robin CPU Scheduling algorithm having better average waiting time (AWT) and average turnaround time (ATT) as compared to other CPU Scheduling techniques. The primary objective of this algorithm is to optimize system performance according to the criteria deemed most important by system designers.**

***Keywords*- First Come First Serve Algorithm, Shortest Job First Algorithm, Round Robin Algorithm, Priority Algorithm, Average Waiting Time, Turnaround Time, Response Time, Throughput.**

I. INTRODUCTION

Operating system performs variety of tasks in which scheduling is one of the basic task. All the resources of Computer are scheduled before use. Scheduling is a technique to instructing a processor, for the order in which it should execute a given set of Processes. Process is a smallest work step of a program which requires a set of resources for its execution that are allotted to it by the CPU [1]. When more than one process are keep coming in the queue one after the other is ready to take control of CPU. In order to execute them in a particular fashion, different scheduling techniques are employed that enable faster and efficient process execution thereby reducing the waiting time faced by each process and increasing CPU utilization.

Throughout its lifetime a process migrates between various scheduling queues by different schedulers until it gets terminated. These queues mainly contain the ready queue which contains set of processes ready for execution. The second queue is the device or the I/O queue which contains all the processes that are waiting for I/O response [1]. The operating system must select processes for scheduling from these queues in a specific manner. This selection process using a particular scheduling technique is carried out by schedulers.

Schedulers in general try to maximize the average performance of a system according to the given criterion [2]. Scheduling Algorithms can be broadly classified into preemptive and non-preemptive scheduling disciplines. The algorithm proposed in this paper is preemptive in nature and attempts to give fair CPU execution time by focusing on average waiting time and turnaround time of a process.

II. SCHEDULING PARAMETERS

Different scheduling algorithms have different characteristics which decide selection of processes using different criteria for execution by CPU. The criteria which decide how one algorithm differs from the other are listed below:

A. Processor utilization

It is the fraction of time during which the processor is busy [2], means the processor is not idle.

B. Throughput

It refers to the amount of work completed in a unit of time [2]. That is, the number of user jobs executed in a unit of time. The more the number of jobs, the more work is done by the system.

C. Turnaround Time

It is defined as the time taken to complete a given process [1]. That is, it is the time spent by the process in the system from the time of its submission until its completion by the system.

D. Waiting Time

Scheduling algorithms do not affect the amount of time during which a process executes or does I/O, it affects only the amount of time spend by the process in the ready queue [1]. That is, the amount of time spent in the ready queue by the process a waiting CPU execution.

E. Response Time

While turnaround time includes total time taken by the process from the time of its submission until the time of its completion, response time is the measure of time from the submission of requests until the first response is produced [1]. This response time does not include the time taken to output that response.

III. OVERVIEW OF EXISTING CPU SCHEDULING ALGORITHMS

CPU scheduling algorithms aim at deciding which Processes in the ready queue are to be allotted to the CPU. Some common CPU scheduling algorithms are.

A. First Come First Served (FCFS) Scheduling

It is the simplest CPU scheduling algorithm that permits the execution of the process on the basis of their arrival time means the process having earlier arrival time will be executed first. Once the control of CPU is assigned to the process, it will not leave the CPU until it completes its execution. For small processes this technique is fair but for long processes it is quite unfair Since FCFS does not involve context switching therefore it has minimal overhead. It has low throughput since long processes can keep processor occupied for a long time making small processes suffer. As a result waiting time, turnaround time and response time can be low [4]. A long burst time process can monopolize CPU, even if burst time of other process is too short called convoy effect. Hence throughput is low [5].

B. Shortest Job First (SJF) Scheduling

This algorithm is non-preemptive in nature and permits the processes to execute first that have smaller burst time [7]. If more than one process has same burst time then control of CPU is assigned to them on the basis of First Come First Served. In most system, this algorithm is implemented for maximum throughput. SJF reduces average waiting time of processes as compared to FCFS. It selects the job with the smallest burst time ensuing CPU availability for other processes as soon as the current process reaches its completion. This prevents smaller processes from suffering behind larger processes in the ready queue for a long time [3]. The difficulty of this algorithm is to know length of CPU burst of next process and it is usually unpredictable [6], there is also a problem of starvation in this algorithm because the arrival of processes having short CPU burst prevents processes having long CPU burst to execute.

C. Round Robin (RR) Scheduling

In this algorithm, a small unit of time called time quantum or time slice is assigned to each process. According to that time quantum processes are executed and if time quantum of any process expires before its complete execution, control is forcefully taken from the current process under execution and is transferred to the next in the queue for the same period of time slice. And it is put at the end of the ready queue. Round Robin is preemptive in nature. The outcome of RR algorithm in term of performance depends entirely on the size of time quantum. If the quantum is very large, RR algorithm works the same as the FCFS algorithm. If the quantum is very small, RR algorithm makes the user feels processor sharing between multiple processes very fast. Average waiting time is high because of FCFS policy and context switching [1].

D. Priority Based Scheduling

Priority scheduling executes processes based on their priority which may be assigned by the system or by the user himself [3]. Processes with the high priority are executed first and those with low priorities are executed next. Processes with equal priority values are executed using FCFS approach [1].

IV. PROPOSED SCHEDULING ALGORITHM

In this algorithm we modified the Round robin scheduling algorithm by the choosing the dynamic time quantum.

For this the steps taken are

Step-1: For the first job time quantum is assigned as half of the burst time of this job

Step-2: the time quantum for later job are calculated as The jobs have different Arrival time. Then put the jobs in ascending order by this formula Factor

F=Arrival time + CPU Burst time.

If the jobs have same F value then put them as first come first serve basis.

Step-3: Then the time Quantum is calculated as Mean of jobs burst times

If the jobs are odd then

Mean is middle job burst time.

If the jobs are even then

(Middle job position burst time + (middle job+1) jobs position burst time) / 2

Step-4: The time quantum is assigned to each process and is recalculated taking the remaining burst time in account after each cycle. This procedure goes on until the ready queue is empty.

V. RESULT AND ANALYSIS

The round robin scheduling is best for those processes that want high response time.

In general round robin algorithm the time quantum is fix and all processes Run according to this time quantum value. This dynamic time quantum approach gives better result on the basis of CPU scheduling criteria.

The following result and analysis show the performance of this approach.
There are some processes and their arrival time and burst time corresponding.

Process	Arrival time	Burst time
P0	0	15
P1	2	13
P2	4	8
P3	5	10
P4	1	12
P5	6	26
P6	3	7

Table-1: Process with arrival time and burst time

If the time quantum is taken in Fix RR method is 5.

Algorithm	Time Quantum	Avg. turnaround Time	Avg. waiting Time	Context Switch
Fix RR	5	64.85	51.85	17
Dynamic SJRR	8,12,7	45.28	32.28	8

Table-2: Comparison of Fix RR and Dynamic SJRR

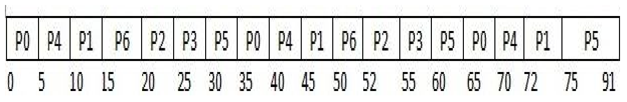

Fig-1: Gantt chart of Fix RR

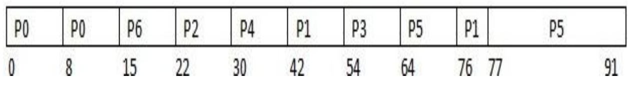

Fig-2: Gantt chart of Dynamic SJRR

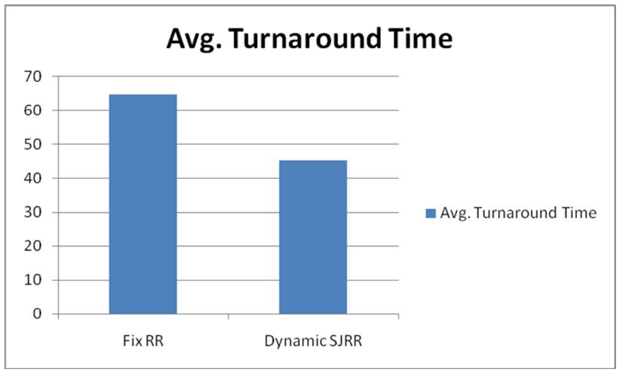

Fig-3: Comparison of Avg. Turnaround time by taking Fix RR and dynamic SJRR

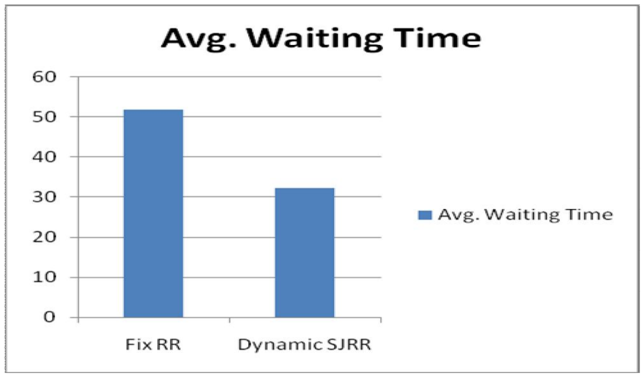

Fig-4: Comparison of Waiting Time by taking Fix RR and dynamic SJRR

By this result it is clear that the proposed method gives better performance.

VI. CONCLUSION

Time quantum is the bottleneck facing round robin algorithm and question arise- What is the optimal time quantum to be used in round robin algorithm? For the effectiveness and the efficiency of the RR algorithm, result of this work provides an answer to this question by using dynamic time quantum instead of fixed time quantum, where the operating system itself finds the optimal time quantum without user intervention. From the comparison of the obtained results, it is observed that proposed algorithm has successfully beaten the existing CPU scheduling algorithms.

It provides good performance in the scheduling policy. As SJF is optimal scheduling algorithm but for large processes, it gives increased waiting time and sometimes long processes will never execute and remain starved. This problem can be overcome by the proposed algorithm.

REFERENCES

[1] Abraham Silberschatz , Peter Baer Galvin, Greg Gagne, "Operating System Concepts",Sixth Edition.

[2] Andrew S. Tanenbaum, Albert S. Woodhull, "Operating Systems Design and Implementation", Second Edition

[3] H.M.Deitel, "Operating Systems", Pearson Education, Second Edition

[4] http://en.wikipedia.org/wiki/Scheduling (computing).

[5] Maj. Umar Saleem Butt and Dr. Muhammad Younus Javed, "Simulation of CPU Scheduling Algorithms",0-7803-6355-8/00/$10.00@2000 IEEE

[6] Rami J. Matarneh, "Self Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes",American Journal of Applied Sciences 6(10): 18311-1837, 2009, ISSN 1546-9239

[7] Gary Nutt, "Operating Systems, A Modern Perspective", Second Edition