International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 08 | Nov-2015 www.irjet.net p-ISSN: 2395-0072

A VARIANT OF ROUND ROBIN ALGORITHM

A. Caroline Mary, S. Joshna

[1] Assistant Professor, Computer Technology, Sri Krishna Arts and Science College, Tamilnadu, Coimbatore
[2] Assistant Professor, Computer Technology, Sri Krishna Arts and Science College, Tamilnadu, Coimbatore

Abstract

CPU Scheduling algorithms help us a long way to allocate jobs to CPU. There are n number of CPU Scheduling algorithms available. These algorithms are rated based on Waiting Time , Turnaround time etc. The ultimate goal for scheduling algorithm is that no jobs should wait for a long time to get in to CPU for processing. User should have a feeling that their work is done in no time. Arrival time of the Job is important consideration, most of all is that the high priority job should be allocated as soon as possible. At times due to the inefficiency of Scheduling algorithm, certain jobs are made to wait for an indefinite time, which we denote as starving. The famous and simplest algorithm being FCFS(First come First Served) concentrates on arrival time of the jobs. Round Robin scheduling algorithm has its own uniqueness, the reason being it gives equal importance to all the process. But here priority of the job is not considered. In this paper basics of Round Robin is maintained with priority in to it. And often switching of jobs is the main drawback of Round Robin which has to be considered. Round Robin comes under pre-emptiveness. That is before the completion of execution of the process it can be removed from CPU. This paper tries to consider that aspect too. Various algorithms are considered for an example, and evaluated to find out the efficiency of Round Robin algorithm with priority.

Key Words: *Scheduling, Waiting Time , Priority,Pre-emptive etc…*

1. INTRODUCTION

Scheduling, ie Process Management is one of the important functionality of an Operating System. Efficient Scheduling Algorithm is needed to avoid indefinite waiting of a Job to be allocated to CPU for execution. Let us find out the efficiency of Scheduling Algorithms and proposed Scheduling Algorithm.

2. SCHEDULING ALGORITHMS

Let us have a look at Scheduling Algorithms available.

Job queue: contains all process in the system

Ready queue: contains all process that reside in main memory and are ready to run

Device queue: contains all process waiting for a particular device

SCHEDULING CRITERIA

1. CPU utilization - CPU Resource should not be in idle state. Maximum usage of CPU is beneficial.
2. Throughput: number of processes that are completed per unit time
3. Turnaround time: interval from the time of submission of a process to the time of completion
4. Waiting time: total amount of time spent by a process in the ready queue

2.1 FCFS (First Come First Served)

Every Job has a arrival Time. The time at which the job entered the scenario to be allocated to CPU. Only the jobs which are at ready state can be allocated to CPU. Ready state meaning, all other resources are allocated except for the Processor. The Job which entered first is assigned to the CPU first. Priority nor the Burse time (Execution Time) is considered. FCFS works under non pre-emptiveness.

1. Maintain a FIFO queue.
2. When a process enters the ready queue, it is placed at the end of the queue.

NON Pre - Emptiveness:

Only after the completion of execution of the process, the process is removed from CPU.

Eg
Table -1:FCFS

PROCESS	ARRIVAL TIME	BURST TIME
P0	2	3
P1	0	5
P2	1	2

Here the arrival of Jobs in the order of P1,P2,P0. Thereby the execution order is also the same p1 - > p2 -> p0.

Waiting Time :
P1 = 0
P2 = 4 (5 - 1)
P0 = 5 (7 -2)
The lesser the waiting time , the greater is the efficiency of Algorithm

2.2 SJF (Shortest Job First)

The drawback with FCFS Algorithm is that at certain scenarios the smallest job (Job with less execution Time) is made to wait for a longer time. To overcome that SJF algorithm is helpful.
When two or more processes waits, SJF favors for the process with smallest burst time ensuring the availability of the CPU when the process reaches the completion.
This prevents smaller processes from suffering behind larger processes in a queue for a long time.

Let us consider an example:

Table -2: SJF

PROCESS	BURST TIME	ARRIVAL TIME
P0	5	0
P1	4	1
P2	2	2

Here the Shortest Job is P1. But at time 0, P0 is the only process available. So no matter what it will be allocated to CPU First.
So the order of Execution is P0 -> P2 -> P1
Waiting Time of P0 = 0
P2 = 3
P1 = 6

Pre-emptive SJF: (shortest remaining time first)

1. When a new process arrives at the ready queue, compare its CPU burst with remaining time for current process.

2.3 Priority Based ALGORITHM

Jobs have different Priorities. According to the importance of Jobs, these priorities are fixed. If 2 process have same priority, then FCFS Algorithm is followed. Example the foreground jobs might have the high priority and so on.

Let us consider an example:

Table -3: Priority Scheduling

PROCESS	BURST TIME	PRIORITY
P0	5	2
P1	4	1
P2	2	0

Here lowest number is considered as the highest Priority. So the order of Execution being P2 - > P1 - > P0
Waiting Time of P2 = 0
P1 = 2
P0 = 6
// Here in this example Arrival Time is same for all the process.

2.4 Round Robin Algorithm

Round Robin Scheduling Algorithm works under pre-emptiveness.

PRE - EMPTIVE:

Even before the completion of execution of the process, the process is removed from CPU and CPU is given to some other process for execution.

Round Robin Algorithm is famous for its unique nature. The main advantage of round robin algorithm over first come first serve algorithm is that it is starvation free . It gives equal importance to all the process. In Round Robin ALGORITHM Time quantum/slice is fixed . Example Time Slice of 2 , 3 etc. Each job is allowed to be in CPU for that quantum of Time. Fairness in execution is made possible in Round Robin Algorithm. No job is made to wait for a longer time. Each Job is given its share of execution time. The only problem with Round Robin Algorithm is the Switch between processes.

Let us consider an example:

Table -4:Round Robin Scheduling

PROCESS	BURST TIME
P0	5
P1	4
P2	2
P3	3

Example : Time Slice = 2
P0 is allocated to CPU for time 2 and then P1 and so on.

P0	P1	P2	P3	
2	2	2	2	(P2 Execution is over- Burst Time = 2)
2	2	0	1	(P1,P3 Execution is over- Burst Time=4,3)
1	0	0	0	(P0 Execution is over- Burst Time = 5)

Waiting Time of Process P0 = 9
P1 = 8
P2 = 4
P3 = 10

Average Waiting Time = (9 + 8 + 4 + 10) / 4 = 7.75

3. ROUND ROBIN WITH PRIORITY

In Round Robin Algorithm, no job is made to wait for indefinite time, but still when high priority Job is not given its importance, we might not get expected results.
Let us consider an example

Table -5 :Round Robin Scheduling with Priority

PROCESS	PRIORITY	BURST TIME
P0	1	5
P1	2	4
P2	3	2
P3	1	3

In this example, less number is treated as a higher priority. Priority in the basis of High - 1, Medium - 2, Low - 3.

So the order of Priority is P0,P3 -> P1 -> P2
Let us consider a Time Slice of 3 for high priority Process. 2 for Medium Priority Process and 1 for Low Priority Process.

Table -6:Round Robin Allocation

P0	P1	P2	P3
3	2	1	3
2	2	1	0

Waiting Time for P0 = 6
P1 = 9
P2 = 12
P3 = 6

Here Waiting Time of High priority process is less as well as no job is made to wait for indefinite time. Even the Job with less priority is given its amount of time, to run into completion.

The high priority process can run into completion sooner thus this algorithm which holds the benefit of both Round Robin and Priority Algorithm is much better.

But the problem with Round Robin Algorithm is the frequent switch between process.

3.1 To Reduce Number of Process Switch

Let us Consider an Example

Table -7:Round Robin Schedule - Process Switch

PROCESS	PRIORITY	BURST TIME
P0	1	5
P1	2	4
P2	3	3
P3	1	5

Time Slice: High Priority Process = 3 , Medium Priority = 2, Low Priority = 1.

Table -8:Round Robin Allocation

P0	P1	P2	P3
3	2	1	3
2	2	1	2
0	0	1	0

Number of Process Switch = 9

If the Remaining Burst Time is very less for the process, the Time Slice can be slightly adjusted. Thereby the number of Process Switch can be reduced.

Let us consider adjustable Time Slice as 1. ie Process P2 at round 2 is in need of 1 more execution Time to complete its execution . To avoid number of Process Switches, the Adjustable Time Slice can be made use of.

Table -9:Round Robin Allocation - Process Switch

P0	P1	P2	P3
3	2	1	3
2	2	2	2

Here the Number of Process Switches = 8.
Thus we can reduce the number of Process Switches at least to some extent by considering Adjustable Time Slice.

4. CONCLUSIONS

One of the main Functionality of Operating System is Process Management. When N number of process are available in the ready queue o be allocated to CPU, it is essential to use best scheduling Algorithm, thereby the waiting time of Jobs can be minimized. Here in this paper, Round Robin with priority in consideration is considered and an adjustable Time Slice is suggested to reduce the number of process switches.

ACKNOWLEDGEMENT

I would like to thank Operating System Staffs of PSG College of Technology.

REFERENCES

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

[2] E.O. Oyetunji, A. E. Oluleye," Performance Assessment of Some CPU Scheduling Algorithms", Research Journal of Information Technology,1(1): pp 22-26, 2009

[3] Neeraj Kumar, Nirvikar," Performance Improvement Using CPU Scheduling Algorithm-SRT",International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)

BIOGRAPHIES

Ms. A. Caroline Mary
Assistant Professor
Sri Krishna College of Arts and Science
Coimbatore

Ms. S. Joshna
Assistant Professor
Sri Krishna College of Arts and Science
Coimbatore