ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (164-168), Month: July - September 2015, Available at: www.researchpublish.com

Revised Round Robin Scheduling Algorithm

[1]Aakash Gupta, [2]Isha Nagpal

[1] Scholar, PPIMT Hisar, [2] Assistant Professor, CSE Department, PPIMT Hisar, India

Abstract: **CPU scheduling is one of the important concept of operating system. It is a method used to schedule jobs for execution. Shortest job next (SJN), also known as Shortest Job First (SJF) or Shortest Process Next (SPN), is a scheduling policy that selects the waiting process with the smallest execution time to execute next. Starvation is the main disadvantage of SJN. Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all processes without priority (also known as cyclic executive). More context switching, large response time, large waiting time, large turnaround time and less throughput are the main disadvantages. Approached Algorithm removes the disadvantage of shortest job next algorithm (Starvation) and Round robin algorithm (Context Switching). It reduces the average waiting time, response time and increase throughput.**

Keywords: **CPU scheduling, Starvation , Context switching, Gantt chart, Shortest job first scheduling algorithm, Round Robin CPU scheduling algorithm, Turnaround time, Waiting time.**

I. INTRODUCTION

The Round Robin scheduling algorithm has the disadvantage of more context switches and it uses FCFS scheduling for selecting the processes for execution. The disadvantage of SJF scheduling algorithm is starvation. So in this paper we try to reduce the disadvantage of both this algorithm and a new approach is used to reduce the number of context switches and problem of starvation.

Features in a good scheduling algorithm

* Minimum turnaround time
* Minimum waiting time
* Minimum context switching
* Maximum CPU utilization
* Maximum throughput

II. PREVIOUS WORK DONE

Various scheduling algorithms already in use and terms used in scheduling are:-

Shortest Job First Scheduling:

This algorithm associates each process the length of the next CPU burst. Shortest-job-first scheduling is also called as shortest process next (SPN). The process with the shortest expected processing time is selected for execution, among the available processes in the ready queue. If the next CPU bursts of two processes are same then FCFS scheduling is used to break the tie.[6]

Starvation:

SJF scheduling leaves the longer jobs waiting indefinitely for the CPU. This is called starvation.[1]

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (164-168), Month: July - September 2015, Available at: www.researchpublish.com

Round Robin Scheduling:

This algorithm is designed for time sharing systems. In this preemption is added with FCFS scheduling to enable the system to switch between processes.

Context Switching:

It is a task of performing a state save of the current process and a state restore of a different process. The context is represented in the PCB of the process; it includes the value of CPU registers, the process state and memory management information.[1]

Waiting Time:

How much time processes spend in the ready queue waiting their turn to get on the CPU. It affects only the amount of time that a process spends to wait for the CPU.[7]

Turnaround Time:

Time required for a particular process to complete, from submission time to completion. Turnaround time is the sum of periods spent waiting to get in to memory, waiting in the ready queue, executing on the CPU and doing input output.[7]

Throughput:

Throughput is the rate at which processes are completed per unit of time.[6]

III. ROUND ROBIN ALGORITHM

Round robin algorithm is simple and easy to implement. The name round robin comes from the principle known as round robin in which every person takes equal share of something in turn[12].

Pseudo Code:
* CPU scheduler picks the process from the circular/ready queue , set a timer to interrupt it after 1 time slice / quantum and dispatches it[2].
* If process has burst time less than 1 time slice/quantum

> Process will leave the CPU after the completion

> CPU will proceed with the next process in the ready queue / circular queue.
Else If process has burst time longer than 1 time slice/quantums > Timer will be stopped. It causes interruption to the OS.
> Executed process is then placed at the tail of the circular / ready queue by applying the context switch.

> CPU scheduler then proceeds by selecting the next process in the ready queue.

IV. PROPOSED ALGORITHM

This proposed algorithm is based on the small change in round robin scheduling algorithm. It selects the job according to the burst time. It uses the SJF scheduling algorithm to select the jobs for execution instead of FCFS algorithm. SJF algorithm's performance is not efficient for the processes having longer CPU burst which increases the waiting time sometimes lead to starvation and also decrease the throughput. So to eliminate this drawback small changes are done.

The new proposed algorithm selects the time quantum equal to the average of the burst time of all the jobs. The proposed algorithm has less number of context switches[8] than Round Robin and less waiting time as compared to SJF algorithm.

Step 1: Firstly the processors are arranged in increasing order according to their CPU burst time.

Step 2: Average of burst time of all the processes is calculated and the time slice is set equal to the average.

Step 3: The shortest process will get the CPU and if the burst time of this process is less than the average then it will leave the CPU for the next process. If the burst time is greater then it will go to the end of ready queue for its turn for the CPU.

Step 4: All the processes are executed in the sequence following the Round Robin scheduling and Shortest Job First scheduling time.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (164-168), Month: July - September 2015, Available at: www.researchpublish.com

EXAMPLE

Let us take 4 jobs for scheduling. Their burst time is given below:-[10]

Jobs	Burst time(ms)
J1	5
J2	8
J3	20
J4	3

According to First in first out scheduling algorithm:-

Gantt chart:-

J1	J2	J3	J4

0 5 13 33 36

jobs	Waiting Time	Turnaround time
J1	0	5
J2	5	13
J3	18	33
J4	26	36

Total waiting time= 0+5+18+26=49ms.

Average waiting time=49/4=12.25ms

Total turnaround time=5+13+33+36=87ms

Average turnaround time= 87/4=21.75ms

According to Round robin scheduling algorithm:-

Time slice=3ms

Gantt chart:-

J1	J2	J3	J4	J1	J2	J3	J2	J3	J2	J3	J3	J3

0 3 6 9 12 14 17 20 22 25 28 31 34 36

Jobs	Waiting Time	Turnaround Time
J1	9	14
J2	14	28
J3	19	36
J4	9	12s

Total Waiting Time = 9+14+19+9=51ms

Average Waiting Time = (Total Wait Time) / (Total number of processes) = 51/4=12.75ms

Total Turnaround Time = 14+28+36+12=90 ms

Average Turnaround Time = (Total Turnaround Time) / (Total number of processes) = 90/4=22.5ms.

Throughput = 4 jobs/36ms = 0.1 jobs/ms

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (164-168), Month: July - September 2015, Available at: www.researchpublish.com

Processes in increasing order

Jobs	Burst time(ms)
J4	3
J1	5
J2	8
J3	20

According to Shortest Job First Algorithm:-[11]

Gantt chart:-

J4	J1	J2	J3

0 3 8 16 36

Jobs	Waiting Time	Turnaround Time
J1	0	3
J2	3	8
J3	8	16
J4	16	36

Total Waiting Time = 0+3+8+16=27ms

Average Waiting Time= (Total Wait Time) / (Total number of processes) = 27/4=6.75ms

Total Turnaround Time = 3+8+16+36=63ms

Average Turnaround Time = (Total Turnaround Time) / (Total number of processes) = 63/4=15.75ms

Throughput = 4 jobs/36ms = 0.1 jobs/ms

According to Proposed Algorithm:-

Total time= 5+8+20+3=36ms

Average time= 36/4=9ms.

Time slice= 9ms.

Gantt chart:-

J4	J1	J2	J3	J3	J3

0 3 8 16 25 34 36

Jobs	Waiting Time	Turnaround Time
J1	0	3
J2	3	8
J3	8	16
J4	16	36

Total Waiting Time = 0+3+8+16=27ms

Average Waiting Time = (Total Wait Time) / (Total number of processes) = 27/4=6.75ms

Total Turnaround Time = 3+8+16+36=63ms

Average Turnaround Time = (Total Turnaround Time) / (Total number of processes) = 63/4=15.75ms

Throughput = 4 jobs/36ms = 0.1 jobs/ms

V. CONCLUSION

The paper presents a new CPU scheduling algorithm which implements the round robin algorithm with some changes. The proposed algorithm follows the shortest job first to select the process for the round robin algorithm. Comparison to the round robin algorithm with the proposed algorithm is done. It is concluded that the proposed algorithm is more efficient than the round robin and shortest job first algorithms as in this the number of context switches is less than the round robin and starvation is reduced as in case of SJF algorithm.

REFERENCES

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

[2] http://javahungry.blogspot.com/2013/09/round-robin-scheduling-algorithm-with-example-java-program-code.html

[3] Jump up^ Arpaci-Dusseau, Remzi H.; Arpaci-Dusseau, Andrea C. (2014), Operating Systems: Three Easy Pieces [Chapter Scheduling Introduction], Arpaci-Dusseau Books

[4] Jump up^ Silberschatz, A.; Galvin, P.B.; Gagne, G. (2005). Operating Systems Concepts (7th ed.). Wiley.p. 161. ISBN 0-471-69466-5.

[5] Jump up^ Tanenbaum, A. S. (2008). Modern Operating Systems (3rd ed.). Pearson Education, Inc. p. 156. ISBN 0-13-600663-9.

[6] http://www.go4expert.com/articles/types-of-scheduling-t22307/.

[7] http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_ CPU_Scheduling.html.

[8] http://whatis.techtarget.com/definition/context-switch.

[9] http://www.researchgate.net/post/What_would_be_the_advantages_and_disadvantages_of_using_Dynamic_Time_slicing_concept_for_scheduling.

[10] http://people.csail.mit.edu/rinard/osnotes/h6.html.

[11] http://en.wikibooks.org/wiki/Operating_System_Design/Scheduling_Processes/SPN.

[12] http://choices.cs.uiuc.edu/~f-kon/RoundRobin/node1.html.