ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 2, pp: (198-203), Month: April - June 2017, Available at: www.researchpublish.com

Median Based Round Robin CPU Scheduling Algorithm

[1]Dolly Khokhar, [2]Mr. Ankur Kaushik

[1]Pursuing Master's in Computer Science and Engineering Meerut Institute of Engineering and Technology (U.P.T.U) Meerut, India
[2]Assistant Professor, Department of IT Meerut Institute of Engineering and Technology Meerut, India

***Abstract*: The essential and important aspect of operating system is multiprogramming. Multiprogramming is a process or method of executing multiple processes simultaneously in the memory. Its main aim to minimize the average waiting time, average turnaround time and maximize the CPU utilization. There are various CPU scheduling algorithms are used to performed multiprogramming like First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling (PS) and Round Robin(RR).The most widely used scheduling algorithm is Round robin scheduling among all of them. It is an optimal CPU scheduling algorithms in timeshared systems.**

This paper presents a new approach that will help to reduce the average waiting time and turnaround time. The proposed algorithm is also compared with various variants of Round Robin algorithm. It produces minimal average waiting time and average turnaround time. In this approach we use mean and median to calculate the time quantum which will improve the effectiveness of the algorithm.

***Keywords:* Context switching, CPU scheduling, Gantt chart, Round Robin CPU scheduling algorithm, Turnaround time, Waiting time, Context Switches.**

I. INTRODUCTION

Modern operating systems support multitasking environment in which processes run in a concurrent manner. In a single-processor system, only one process can run in the CPU at a time. Others processes in the ready queue must wait until the CPU becomes free. The operating system must decide through the scheduler the order of execution of the processes in ready state. The objective of multiprogramming is to have some process running at all times to maximize CPU utilization. Scheduling is a fundamental operating-system function. Almost all computer resources are scheduled before using. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating-system design. CPU scheduling determines which processes run when there are multiple run-able processes. CPU scheduling is important because it can have a. big effect on resource utilization and the overall performance of the system. In general we want to optimize the behaviour of the system.

II. SCHEDULING CRITERIA

The selection criteria of a CPU scheduling algorithm depend upon the following [1]:

1) Fairness: All processes must fairly get the CPU and no one gets into starvation.

2) CPU utilization: CPU should remain busy for 100% time.

3) Throughput: Increase the number of processes that have finished their execution within a certain time interval.

4) Response time: It is the time when request is submitted for the process till the first response of the process is produced.

5) Waiting time: It is the time a process spends in ready queue.

6) Turnaround time: It is the time from submission of the request till the time it is completed.

7) Context Switch: When a process is preempted, its context is stored so that it can resume later from the same point. It is totally an overhead because CPU does no useful work during context switch. Also it adds overhead for the scheduler.

So we can conclude that a good scheduling algorithm for real time and time sharing system must possess following characteristics:

- Minimum context switches.
- Maximum CPU utilization.
- Maximum throughput.
- Minimum turnaround time.
- Minimum waiting time.
- Minimum response time.

III. SCHEDULING ALGORITHM

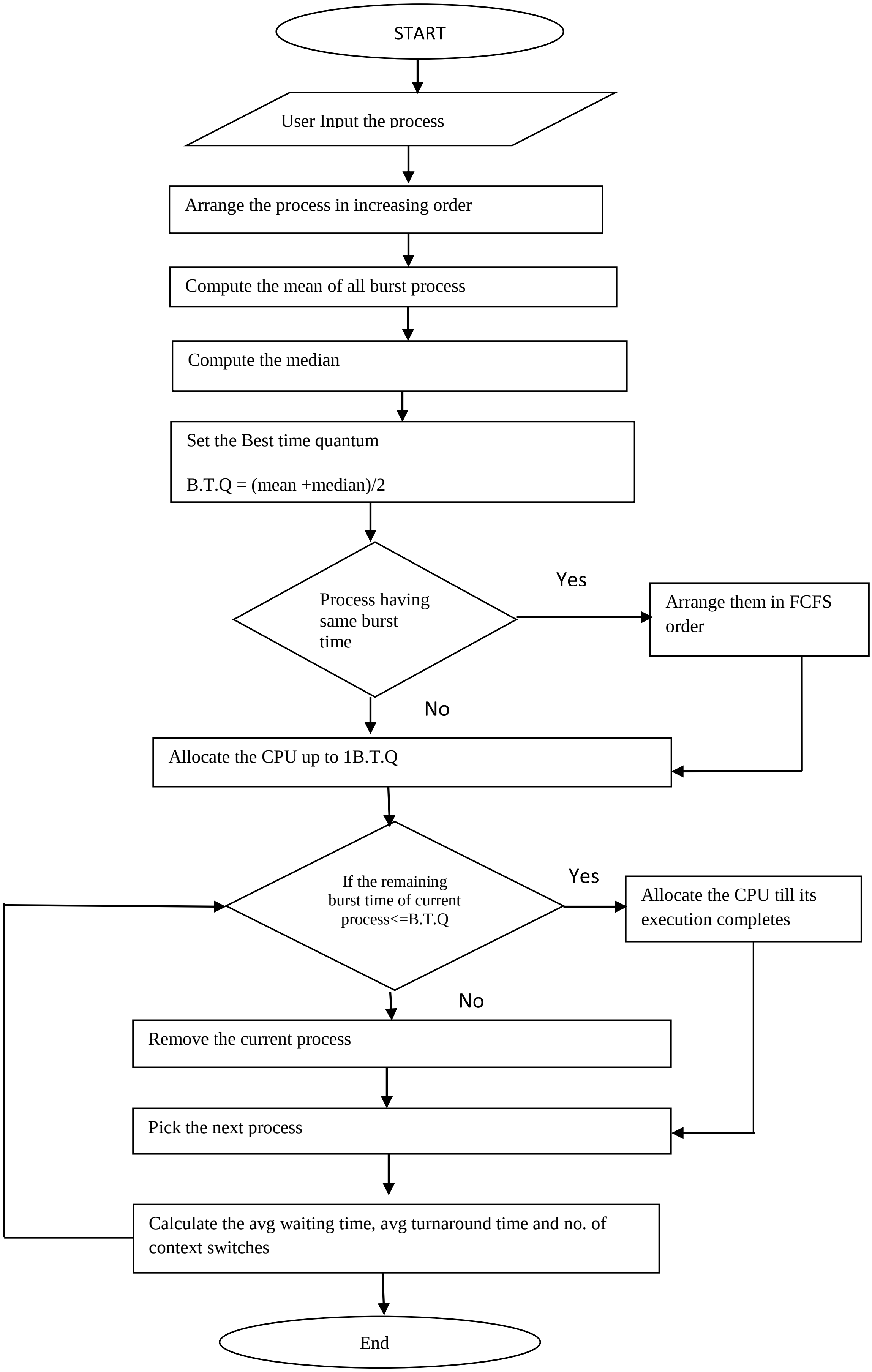
CPU scheduling algorithms are used to allocate the CPU to the processes waiting in the ready queue. First-Come-First-Served (FCFS) Round Robin (RR), Shortest Job First (SJF) and, Priority Scheduling are some popular CPU scheduling algorithms. In FCFS CPU scheduling algorithm, the process that arrives first in the ready queue is served first. The average waiting time in this scheduling is quite long [2]. In SJF CPU scheduling algorithm, the process with shortest CPU burst time executes first from the ready queue. In SJF average waiting time decreases. CPU is allocated to the processes based on their priority in Priority scheduling algorithm. The process with highest priority gets executed first and then the second highest and so on. Each process from the ready queue is given a fixed time quantum in RR CPU scheduling algorithm.

IV. PREVIOUS WORK DONE

Round Robin scheduling algorithm using static time quantum has many drawbacks. In the recent years, many research works has been done to improve the performance of Round Robin algorithm. A Self Adjustment Round Robin [3] scheduling algorithm uses dynamic time quantum repeatedly adjusted according to the burst time of the now-running processes. In SJRR CPU scheduling algorithm[4] the process having minimum burst time is selected first then second one and time quantum become equals to minimum burst time in this way it improves the waiting time. In IRR CPU scheduling algorithm[5] the process is allocated to the CPU only one time as like simple Round Robin algorithm and then the process having shortest job first is allocated to the CPU after its full execution it is removed from the waiting queue and the similarly the next process is executed and so on. In [6] author proposed algorithm (AAIRR) focuses on improving more on the improved Round Robin CPU scheduling algorithm. The algorithm by reduces the waiting time and turnaround time drastically compared to the simple Round Robin scheduling algorithm. This proposed algorithm works in a similar way as but with some modification. It works in three stages: Stage 1: It picks the first process that arrives to the ready queue and allocates the CPU to it for a time interval of up to 1 time quantum. After completion of process's time quantum, it checks the remaining CPU burst time of the currently running process. If the remaining CPU burst time of the currently running process is less or equal to 1 time quantum, the CPU is again allocated to the currently running process for remaining CPU burst time. In this case this process will finish execution and it will be removed from the ready queue. The scheduler then proceeds to the next shortest process in the ready queue. Otherwise, if the remaining CPU burst time of the currently running process is longer than 1 time quantum, the process will be put at the tail of the ready queue. Stage 2: The CPU scheduler will then select the next shortest process in the ready queue, and do the process in stage 1. Stage 3: For the complete execution of all the processes, stage 1 and Stage 2 have to be repeated.

V. PROPOSED ALGORITHM

The proposed algorithm focus on the selection of time quantum. The algorithm by reduces the no of context switch, waiting time and turnaround time drastically compared to the IRR Scheduling algorithm and simple Round Robin scheduling algorithm. In this algorithm we use mean and median to calculate the best time quantum. The proposed algorithm eliminates the discrepancies of implementing simple round robin architecture. In the first stag all the processes are arranged in the increasing order of CPU burst time. Then in the second stage the algorithm calculates the mean of the CPU burst time of all the processes. After calculating the mean, median is calculated .And with the help of mean and median we calculate the time quantum. Then in the last stage algorithm pick the first process from the ready queue and allocate the CPU to the process for a time interval of up to 1 best time quantum. If the remaining burst time of the current running process is less than 1 Best time quantum then algorithm again allocate the CPU to the Current process till it execution. After execution it will remove the terminated process from the ready queue and again go to the stage 3. The flowchart for proposed algorithm is shown below in figure 1:


START
User Input the process
Arrange the process in increasing order
Compute the mean of all burst process
Compute the median
Set the Best time quantum
B.T.Q = (mean +median)/2
Process having same burst time
Yes
Arrange them in FCFS order
No
Allocate the CPU up to 1B.T.Q
If the remaining burst time of current process<=B.T.Q
Yes
Allocate the CPU till its execution completes
No
Remove the current process
Pick the next process
Calculate the avg waiting time, avg turnaround time and no. of context switches
End

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 2, pp: (198-203), Month: April - June 2017, Available at: www.researchpublish.com

STEPS FOR MODIFIED ROUND ROBIN ALGORITHM

Step 1 - Start

Step 2 - Arrange the process in increasing order of the CPU burst time in ready queue.

Step 3 - Calculate the mean of the CPU burst time of all the process.

mean=(P1+P2+P3+.....Pn)/n

Step 4 - Calculate the median .

Step 5 - Calculate the best time quantum

B.T.Q= [mean + median]/2

Step 6 - Allocate the CPU to that process that have less arrival time .

(a)- If any two process have same arrival time allocate the CPU to that process that have less burst time.

(b)- After that compare the burst time of all the process to the current process if two or more than two process have less burst time then check their arrival time and execute the current process upto the less arrival time of the current process and then allocate the CPU to the process that have less arrival time.

(c)- If the current processes have lowest burst time then execute it upto B.T.Q otherwise go to step 6(a).

Step 7 - If the remaining CPU burst time of the current process is less than or equal to B.T.Q then

(a)- Re allocate the CPU to the current process again for the remaining burst time. After the complete execution of the current process remove it from ready queue.

(b)- Otherwise remove the process from the ready queue and put it on the tail of the ready queue for further execution.

Step 8 - Pick the next process from the ready queue and allocate the CPU to it upto the B.T.Q and go to step 7 .

Step 9 - If the ready queue is empty then go to step 11.

Step 10 - If there is any process left in the ready queue allocate the CPU to that process otherwise go to step 3.

Step 11 - Calculate the average waiting time (AWT) , average turnaround time (ATAT), and number of context switches.

For example:

A ready queue with four processes P1, P2, P3 and P4 has been considered for illustration purpose.

The processes are arriving at time 0 with burst time 12, 8, 21 and 15 respectively. The processes

P1, P2, P3 and P4 are arranged in the ascending order of their burst time in the ready queue which gives the sequence P2, P1, P4 and P3.

Now the mean of the CPU burst time is calculated i.e 14.After that the median is calculated i.e 14.Now we use the mean and median to calculate the best time quantum. The time quantum value is equal to 14. CPU is allocated to the processes P2, P1, P4 and P3 from the ready queue for a time quantum 14.The average waiting time is15.75 and average turnaround time is 14.

Grantt Chart of AAIRR Scheduling Algorithm:

P1	P1	P1	P2	P2	P3

0 5 10 12 17 20 25

P3	P3	P3	P3	P4	P4	P4

30 35 40 41 46 51 56

Grantt Chart of AIRR Scheduling Algorithm:

P2	P1	P4	P3	P1

0 8 16 24 32 36

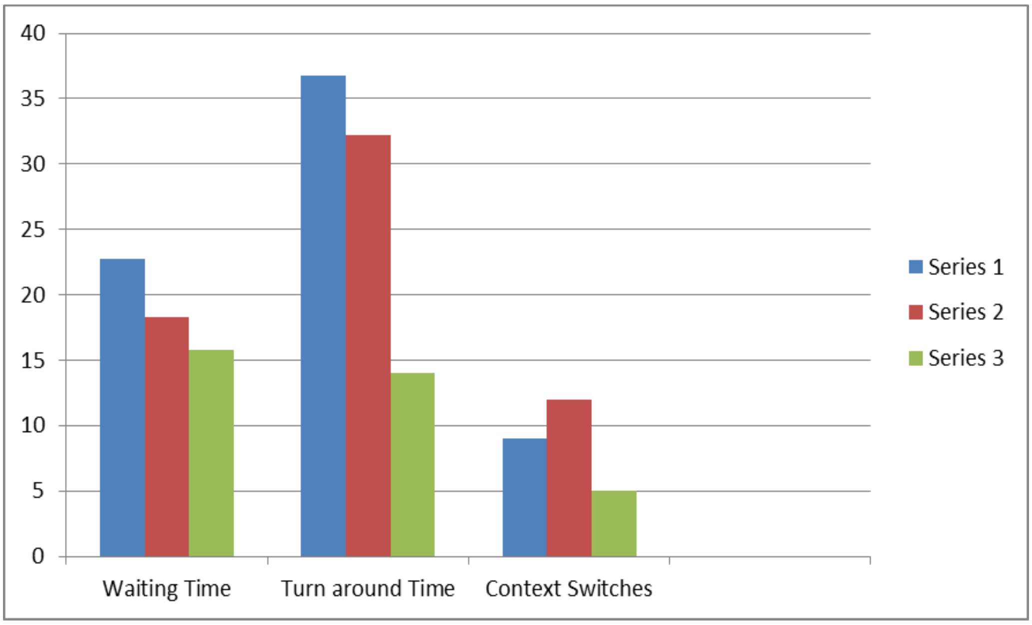
P4	P3	P4	P3	P3

36 40 44 47 50 56

Grantt Chart Of Proposed Algorithm:

P2	P1	P4	P4	P3	P3

0 8 20 34 35 49 56



VI. CONCLUSION

In this paper, we have presented a Median Based Round Robin CPU scheduling algorithm which improved on the An Advanced Improved Round Robin Scheduling algorithm [7] and Additional Improvement on Round Robin CPU scheduling algorithm [6]. Results have shown that the proposed algorithm gives better results in terms of average waiting time, average turnaround time and number of context switches in all cases of process categories than the simple Round Robin CPU scheduling algorithm, Improved Round Robin CPU scheduling algorithm and the An Additional Improvement on Round Robin CPU Scheduling algorithm. The general problem in any form of Round Robin CPU scheduling algorithm, Improved Round Robin CPU scheduling algorithm and the An Additional Improvement Round Robin CPU Scheduling algorithm is the choice of time quantum.

REFERENCES

[1] Arfa Yasin, Ahmed Faraz, Saad Rehman, Prioritized Fair Round Robin Algorithm with Variable Time Quantum, *13th International Conference on Frontiers of Information Technology 2015 IEEE*

[2] A. Silberschatz, P. B. Galvin & G. Gagne, (2005) Operating System Concepts, John Wiley and Sons Inc, pp 157-167.

[3] 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): 1831-1837, 2009, ISSN 1546-9239 © 2009 Science Publications*

[4] Rakesh patel , SJRR CPU Scheduling Algorithm*IJECS Volume 2 Issue 12, Dec. 2013, Page No.3396-3399*

[5] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash, Himanshu Sharma," An Improved Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1064-1066, 2010

[6] Abdulraza, Abdulrahim,Salisu Aliyu, Ahmad M Mustapha, Saleh E Abdullahi, " An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 2, February 2014 ISSN: 2277 128X.

[7] http://en.wikipedia.org/wiki/Scheduling

[8] Manish Kumar Mishra1 and Dr. Faizur Rashid AN IMPROVED ROUND ROBIN CPU SCHEDULING ALGORITHM WITH VARYING TIME QUANTUM International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.4, No.4, August 2014