International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-1 Issue-3, August 2012

Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum

Pallab Banerjee, Probal Banerjee, Shweta Sonali Dhal

***Abstract:- Round Robin Scheduling algorithm is designed especially for time sharing Operating system (OS).It is a preemptive CPU scheduling algorithm which switches between the processes when static time Quantum expires. The Round Robin Scheduling algorithm has its disadvantages that is its longer average waiting time, higher context switches, higher turnaround time .In this paper a new algorithm is presented called Average Max Round Robin (AMRR) scheduling algorithm .In this scheduling algorithm the main idea is to adjust the time Quantum dynamically so that (AMRR) perform better performance than simple Round Robin scheduling algorithm*.**

Keywords-Operating System, Round Robin, Average Max Round Robin, Turnaround time, Waiting time, Context Switch.

I. INTRODUCTION

An operating system is a system software which makes an interface between user and computer hardware. Operating system provides a platform in which user can interact with hardware and execute programs in an efficient manner. Modern operating system and time sharing system are more complex, they have evolved from a single task to multitasking environment in which processes run in synchronized manner. In a multiprocessing and multitasking environment if several processes are ready to run at the same time, the system must choose among them and assigned to run on the available CPUs, is called CPU scheduling. Allocating CPU to a process requires careful awareness to assure justice and avoid process starvation for CPU. Scheduling decision try to reduce the following: turnaround time, response time and average waiting time for processes and number of context switches [4]. CPU scheduling algorithm decides which of the processes in the Ready Queue(RQ) are to be allocated to the CPU.

Revised Manuscript Received on 30 August 2012

*Correspondence Author(s)

Pallab Banerjee, Lecturer, CSE Department, Cambridge Institute of Technology, Ranchi.

Probal Banerjee, Lecturer, ECE Department, Cambridge Institute of Technology, Ranchi.

Shweta Sonali Dhal, Lecturer, EEE Department, Cambridge Institute of Technology,Ranchi.

There are many different CPU scheduling algorithms used like FCFS, SJF, RR, Priority scheduling algorithm and Short Remaining Time Next (STRN) Remaining Time Next (STRN) algorithm.The processes are scheduled according to the given burst time, arrival time, time quantum and priority. Out of those algorithms, Round Robin (RR) is the oldest, simplest and most widely used proportional share scheduling algorithm. It is similar to FCFS scheduling, but preemption is added to switch between processes. In Round Robin algorithm a small unit of time slice are required which is called Time Quantum (TQ). The CPU scheduler goes around Ready Queue and allocates the CPU to each processes by the help of Dispatcher for a time interval of up to 1 Time Quantum(TQ)[1]. If new process arrives then it is added to the tail of Circular Queue. The CPU scheduler picks the first process from the Ready Queue sets a timer to interrupt after one Time Quantum and dispatches the process[8].After TQ is expired , the CPU preempts the process and the process is added to the tail of the Circular Queue. If the process finishes before the end of the TQ, the process itself preempts the CPU willingly[3].In this paper, we tried to solved the Time Quantum problem by adjusting the Time Quantum Dynamically with respect to the existed set of processes in Ready Queue

II. PRELIMINARIES

Program is refer to the set of instruction that are executed in pipeline fashion Program in execution is called process. Process are represented by Process Control Block (PCB).PCB contains many information about process such as process state, process number, program counter, list of open files, registers and CPU scheduling information[4]. When process enter into the main memory and are ready and waiting to execute are kept in the data structure called Ready Queue. When a process assign to the CPU, it execute or while waiting for some event to occure. The process which are waiting for I/O request are kept in Device Queue. The Long term scheduler or job scheduler select process from job pool and load them into main memory for execution. Short term scheduler or CPU scheduler select from among the processes that are ready to execute and allocates the CPU to one of them.

*Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)*

Medium term scheduler is used in time sharing system. The main advantage of Medium term scheduler is sometimes it remove processes from main memory and thus reduce degree of multiprogramming. Later the process can be reintroduced into memory, and its execution can be continued where it left off. This scheme is called as swapping. So, the process is swapped out, and is later swapped in, by the medium term scheduler [1].

The scheduler is mainly concerned with:

Arrival time:-The time at which process arrives in main memory.

Burst time: - The time for which a process holds the CPU.

Waiting time: - The amount of time, process waiting in the Ready Queue.

Turnaround time: - The total time between arrival of a process and process completion.

Response time: - The time from the submission of a request by a process till its first response.

III. RELATED WORK

In the last few years different approaches are used to increase the performance of Round Robin scheduling like Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice [4], Multi-Dynamic time Quantum Round Robin (MDTQRR) [5]. Min-Max Round Robin (MMRR) [3], Self-Adjustment Time Quantum in Round Robin (SARR) [8], Dynamic Quantum with Re-adjusted Round Robin (DQRRR)[11].

IV. PROPOSED APPROACH

Let's assume that the burst time of the processes is taken as sorted increasing order so that it will give better turnaround time and waiting time. Generally in Round Robin algorithm the performance depends upon the size of fixed or static Time Quantum (TQ). If TQ is too large then Round Robin algorithm approximate to First Come First Served (FCFS). If the Time Quantum is too small then there will be many context switching between the processes. So, our approach solved this problem by taking a dynamic TQ. Where TQ is the mean of the summation of the average and the Maximum Burst time.

AVG = (Summation of Burst Time of all the processes)/ Number of processes

TQ=(AVG+MAXBT)/2

V. PROPOSED ALGORITHM

In our proposed algorithm, processes are already present in the Ready Queue (RQ). By default, Arrival Time (AT) is assigned to zero. The number of processes 'n' and CPU Burst Time (BT) are accepted as input and Average Turnaround Time (ATT), Average Waiting Time (AWT) and number of Context Switch (CS) are produced as output. Let TQ and TQn be the time quantum and new time quantum respectively. The pseudo code for the algorithm is presented in Figure 1 and the flowchart of the algorithm is presented in Figure 2.

VI. PERFORMANCE METRICS

The proposed algorithm is designed to meet all scheduling criteria such as maximum CPU utilization, maximum throughput, minimum turnaround time, minimum waiting time and context switches. Here we are considering three performance criteria in each case of our experiment.

1. **Turnaround Time(TAT)**

   TAT=Finish Time –Arrival Time

   Average Turnaround Time should be less.

2. **Waiting Time(WT)**

WT= Start Time- Arrival Time

Average Waiting Time should be less.

3. **Context Switch**

The number of context Switch should be less

VII. ILLUSTRATION

Suppose four processes arriving time =0,and CPU burst time is (P1=70, P2=65,P3=10, P4=15).Then the processes are sorted in ascending order which results in sequence P3=10,P4=15,P2=65 ,P1=70. Then TQ is calculated. Where TQ is the mean of the summation of the average and the Maximum Burst time i.e. AVG=(10+15+65+70)/4=40. So TQ is equal to (AVG+MAXBT)/2 i.e. TQ=(40+70)/2=55. After first iteration the remaining CPU burst time sequence is P3=0,P4=0,P2=10,P4=15.In this case, processes P3 and P4 are deleted from the Ready Queue. Again CPU burst time is sorted in ascending order and new TQ is calculated. Here new TQ is equal to 14. After second iteration the remaining CPU burst time sequence is P2=0 and P4=1. Then P2 is deleted from RQ. After third iteration the remaining CPU burst time sequence is P4=0. Since, now there is no process in the RQ, it completes its execution and ATT,AWT and CS are calculated. In this case, ATT=98.75,AWT=58.75,CS=5.

VIII. EXPERIMAENTAL ANALYSIS

In every case we will compare the result of the proposed AMRR method with Round Robin scheduling algorithm. Here we have taken 20 as the static time quantum (TQ) for RR algorithm

CASE 1:-

Let's consider four processes(P1,P2,P3,P4) with arrival time =0 and burst time(8,40,72,84) as shown in the Table 1.Table 2 shows the output using RR algorithm and AMRR algorithm. Figure 3 and Figure 4 shows Gantt chart of both RR and AMRR algorithm respectively.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

```
1.  All the process in the ready Queue are stored in
    ascending order
    // n= Number of processes
    // i= Loop variable (counter)
    //BT= Burst Time
2.  while(RQ!=NULL)
    // RQ=Ready Queue
    //TQ=Time Quantum
    //AVG=Average of all the processes
    //MAXBT=Maximum Burst Time
    AVG= (Sum of BT of all the process)/Number of
    processes.
    //(Use round off function in AVG.)
    TQ=(AVG+MAXBT)/2
    //(Use round off function in TQ)
     (Remaining Burst time of the process)
    //If one process is there then TQ is equal to BT
    itself
3.  // Assign TQ to (1 to n) processes
    for i=0 to n loop
    {
    Pi->TQn
    }
    //TQn=New Time Quantum
    end of for
    //Assign TQn to all the available processes
4.  Calculate the remaining Burst time of the
    processes.
5.  If (new process arrived and BT!=0 or new
    process is arrived and BT==0)
    then go to step1
    else if (new process is not arrived and BT!=0)
    then go to step 2
    else
    go to step 6
    end of if
    end of while
6.  Calculate ATT,AWT,CS
    //ATT=Average Turnaround time
    //AWT=Average waiting time
    //CS=Number of context switch
7.  End
```

Figure 1. Pseudo code for Avg Max Round Robin (AMRR) algorithm

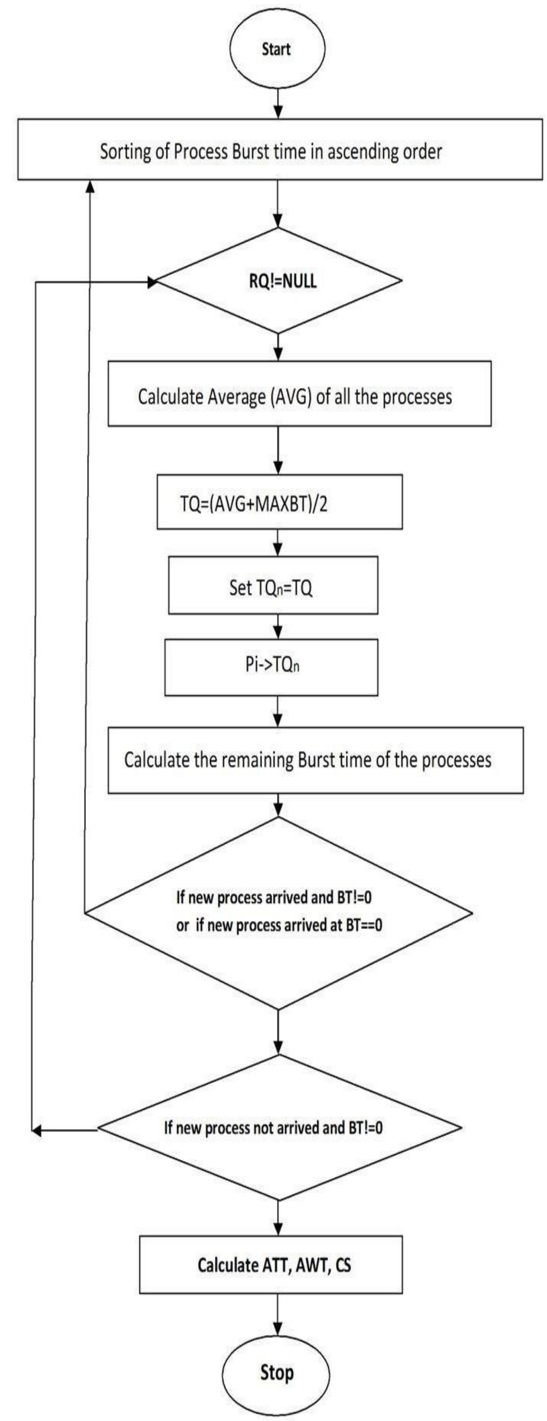


Figure 2. Flowchart of Avg Max Round Robin (AMRR) algorithm

Table 1. Processes with Burst Time

Process	Arrival Time	Burst Time
P1	0	8
P2	0	40
P3	0	72
P4	0	84

Table 2: Comparison between RR algorithm and our new proposed AMRR algorithm (CASE1).

Algorithm	Time Quantum	Turnaround Time	Waiting Time	Context Switch
RR	20	120	69	10
AMRR	68,13,3	112	61	5

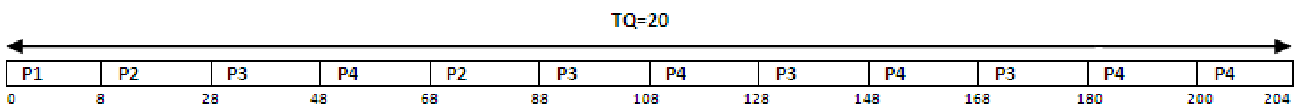


Fig.3: Gantt chart of RR from Table 1(CASE1).

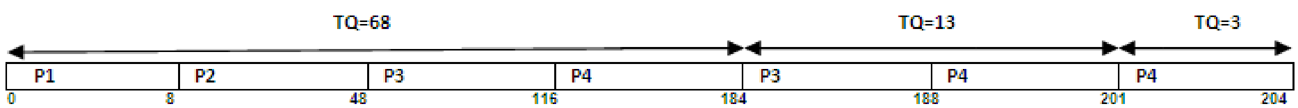


Fig.4: Gantt chart of AMRR from Table 1(CASE 1).

CASE 2:-

Let's consider four processes(P1,P2,P3,P4) with arrival time =0 and burst time(41,42,43,44) as shown in the Table 3. Table 4 shows the output using RR algorithm and AMRR algorithm. Figure 5 and Figure 6 shows Gantt chart of both RR and AMRR algorithm respectively.

Table 3. Processes with Burst Time

Process	Arrival Time	Burst Time
P1	0	41
P2	0	42
P3	0	43
P4	0	44

Table 4: Comparison between RR algorithm and our new proposedAMRR algorithm (CASE 2).

Algorithm	Time Quantum	Turnaround Time	Waiting Time	Context Switch
RR	20	165	122.5	11
AMRR	43,1	105	62.5	3

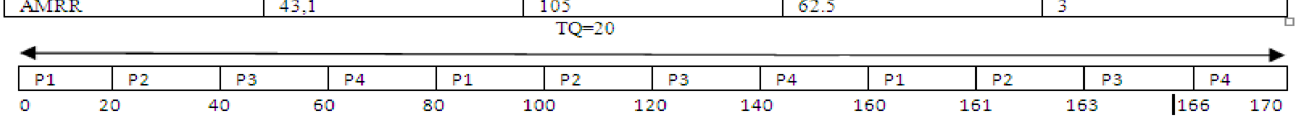


Fig.5: Gantt chart of RR from Table 3(CASE 2)

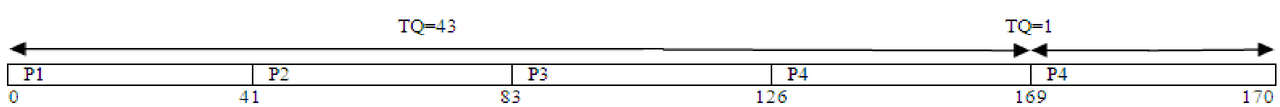


Fig.6: Gantt chart of AMRR from Table 3(CASE 2).

CASE 3:-

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Let's consider four processes(P1,P2,P3,P4) with arrival time =0 and burst time(25,50,180,200) as shown in the Table 5. Table 6 shows the output using RR algorithm and AMRR algorithm. Figure 7 and Figure 8 shows Gantt chart of both RR and AMRR algorithm respectively.

Table 5. Processes with Burst Time

Process	Arrival Time	Burst Time
P1	0	25
P2	0	50
P3	0	180
P4	0	200

Table 6: Comparison between RR algorithm and our new proposed AMRR algorithm (CASE 2).

Algorithm	Time Quantum	Turnaround Time	Waiting Time	Context Switch
RR	20	277.5	173.75	22
AMRR	157,38,5	241.75	128	5

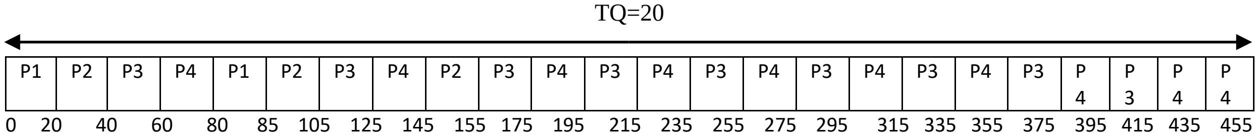


Fig.7: Gantt chart of RR from Table 5(CASE 3)

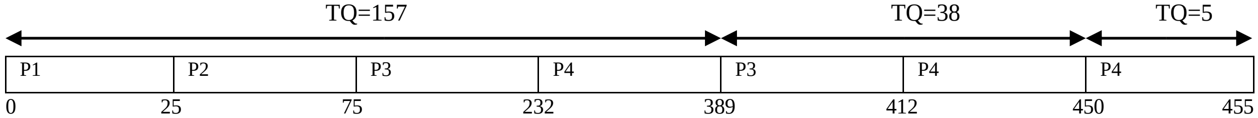


Fig.8: Gantt chart of AMRR from Table 5(CASE 3).

CASE 4:-

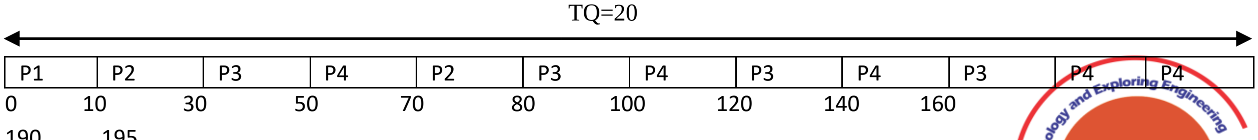
Let's consider four processes(P1,P2,P3,P4) with arrival time (0,4,20,25) and burst time(10,30,70,85) as shown in the Table 7. Table 8 shows the output using RR algorithm and AMRR algorithm. Figure 9 and Figure 10 shows Gantt chart of both RR and AMRR algorithm respectively.

Table 7. Processes with Burst Time

Process	Arrival Time	Burst Time
P1	0	10
P2	4	30
P3	20	70
P4	25	85

Table 8: Comparison between RR algorithm and our new proposed AMRR algorithm (case 4).

Algorithm	Time Quantum	Turnaround Time	Waiting Time	Context Switch
RR	20	101.5	52.75	10
AMRR	67,15,3	93.25	44.5	5



Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Fig.9: Gantt chart for RR in Table 7(case 4)

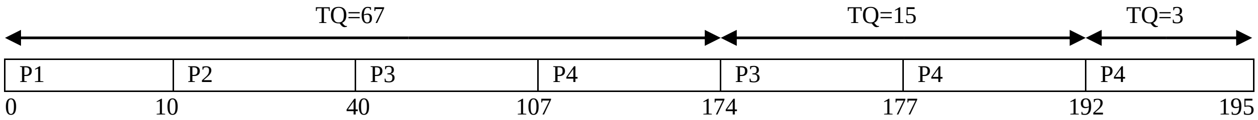


Fig.10: Gantt chart for AMRR in Table 7(case 4).

CASE 5:-

Let's consider four processes(P1,P2,P3,P4) with arrival time (0,18,30,50) and burst time (25,55,80,100) as shown in the Table 9.

Table 10. shows the output using RR algorithm and AMRR algorithm. Figure 11 and Figure 12 shows Gantt chart of both RR and AMRR algorithm respectively.

Table 9. Processes with Burst Time

Process	Arrival Time	Burst Time
P1	0	25
P2	18	55
P3	30	80
P4	50	100

Table 10: Comparison between RR algorithm and our new proposed AMRR algorithm (case 5).

Algorithm	Time Quantum	Turnaround Time	Waiting Time	Context Switch
RR	20	156.75	89.25	11
AMRR	83,17	106.75	41.75	3

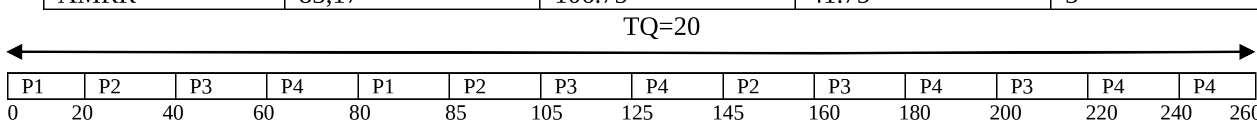


Fig.11: Gantt chart for RR in Table 9(case 5)

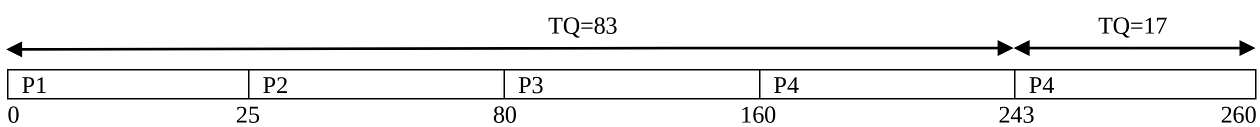


Fig.12: Gantt chart for AMRR in Table 9(case 5).

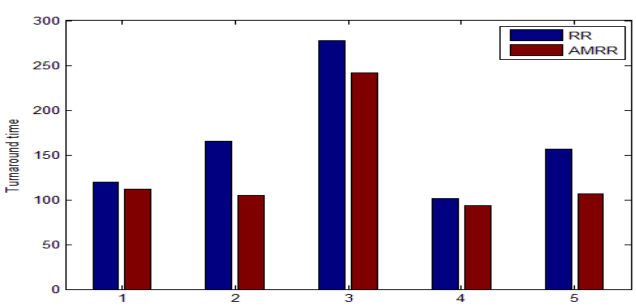


Fig.13: Comparison of average turnaround time of RR and AMRR taking arrival time into consideration.

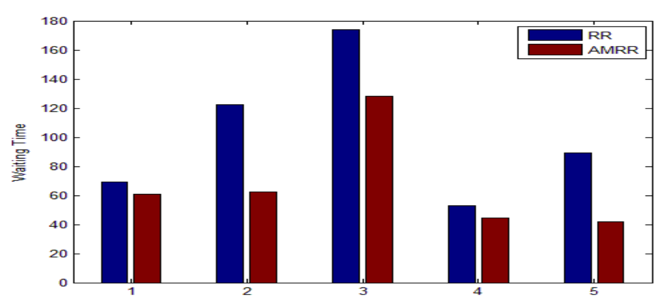


Fig.14. Comparison of average waiting time of RR and AMRR taking arrival time into consideration.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

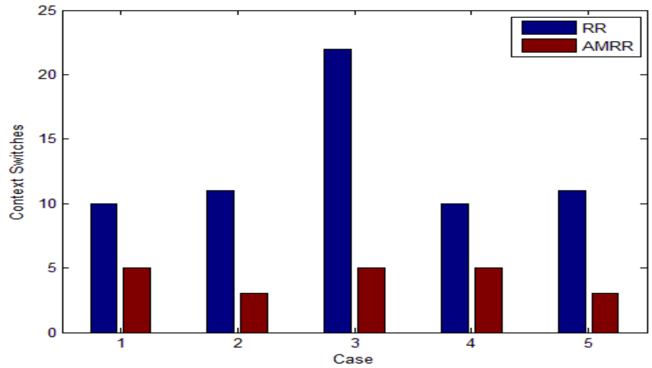


Fig.15. Comparison of Context Switches of RR and AMRR taking arrival time into consideration.

REFERENCES

1. "Silberschatz, A., P.B. Galvin and G. Gagne, 2008" Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA., ISBN: 13: 978-0471694663, pp: 944.
2. "Tanebaun, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13:9780136006633, pp: 1104.
3. Sanjay Kumar Panda and Saurav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure" ISSN : 0975-3397 ,Vol. 4 No. 01, January 2012.
4. Sarojhiranwal and D.r. K.C.Roy"Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice".volume 2,issue 3.
5. H. S. Behera, Rakesh Mohanty, Sabyasachi Sahu and Sourav Kumar Bhoi." Comparative performance analysis of multi-dynamic time quantum round robin (mdtqrr) algorithm with arrival time", ISSN : 0976-5166, Vol. 2, No. 2, Apr-May 2011.
6. 10"Tarek Helmy, Abdelkader Dekdouk" Burst Round Robin: As a Proportional-Share Scheduling Algorithm, IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations, pp. 424-428, 11-14 November,2007
7. "Yaashuwanth .C & R. Ramesh" Inteligent time slice for round robin in real time operating system, IJRRAS 2 (2), February 2010
8. [8] R. J. Matarneh, "Seif-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Proceses", American Journal of Applied Sciences 6 (10), pp. 1831-1837, 2009.
9. J. Nieh, C. Vaill and H. Zhong, "Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler", Proceedings of the USENIX Annual Technical Conference, Boston, Massachusetts, USA, pp. 25-30, June 2001.
10. S. M. Mostafa, S. Z. Rida and S. H. Hamad, "Finding Time Quantum of Round Robin CPU Scheduling Algorithm in General Computing Systems using Integer Programming", IJRRAS 5 (1), pp.64-71, October 2010.
11. H. S. Behera, R. Mohanty, and D. Nayak, "A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis," vol. 5, no. 5, pp. 10-15, August 2010.
12. A. Bhunia, "Enhancing the Performance of Feedback Scheduling", IJCA, vol. 18, no. 4, pp. 11-16, March 2011.
13. "Prof. Rakesh Mohanty, Prof. H. S. Behera, Khusbu Patwari, Manas Ranjan Das, Monisha Dash, Sudhashree" Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm, Am. J. Applied Sci., 6 (10): 1831-1837, 2009.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)