A Varied Round Robin Approach using Harmonic Mean of the Remaining Burst Time of the Processes

Adeeba Jamal
Department of Computer Science
Jamia Hamdard, New Delhi, India

Aiman Zubair
Department of Computer Science
Jamia Hamdard, New Delhi, India

ABSTRACT

A conventional round robin is a distinctive approach to the CPU scheduling algorithm. It is somehow related to the First Come First Serve approach with preemption included to give a fair chance to all the processes to execute waiting in the ready queue. A fixed time period known as time quantum is defined. The predominant round robin is an impartial algorithm since each process is given a fair share to complete its execution on its chance. No process is apportioned the CPU for more than one time quantum, so even if a fraction of time is remaining for a process to conclude its execution, the process is directed back to the ready queue and has to wait for its turn. Here, in this paper we have put forth an approach which will vanquish the challenge which the conventional round robin faces.

KEYWORDS-- harmonic mean; ready queue; time quantum; left over time;

1. INTRODUCTION

CPU Scheduling forms the foundation of the multi-programmed operating systems. In the single processor system, only one process can run at that particular time and the others have to wait until the CPU is idle and then it is rescheduled[16], [17]. But the concept of multi-programming gives emphasis on the idea which says that many processes can run at all times and this maximizes the CPU utilization and the CPU is not left idle[17],[18]. Round robin algorithm is precisely designed for the time sharing systems and follows a circular queue. We define a particular unit of time called time quantum, each process runs to the limit of the time quantum and the new processes are added to the tail of the ready queue [16], [17], [18]. If the process has the burst time within the limit of the time quantum then that process is executed otherwise that process has to wait for its next cycle, this increases the number of context switches, average turnaround time as well as the average wait time [16]. But here in our altered algorithm, we have repeatedly calibrated the time quantum in accordance with the left over time of the processes in the ready queue. According to the experimental analysis of our modified algorithm it can be seen that our algorithm enacts much better than the conventional round robin, by diminishing the average wait time, average turnaround time as well as the number of context switches. In our paper, we have proposed a new compelling algorithm for the round robin scheduling algorithm, which is based on the harmonic mean of the remaining burst time of the processes. We, in our feature are calculating the harmonic mean of the sum of the left over time of the processes, and then the time quantum is altered according to the provision of the algorithm stated.

2. RELATED WORK

Significant additions have been made towards improving the performance of the round robin CPU scheduling algorithm. Various features have put forth their approaches to overcome the limitations of the classical round robin algorithm in the past few years. As in feature [1] the scrivener has used a Min-Max dispersion measure in accordance with remaining CPU burst time to calculate the time quantum for the processes. Similarly in [2] the authors a proposed a method wherein they use the median and the variance to carry out round robin scheduling with a dynamic time quantum. In [3] the writer uses a comparative approach by placing the burst time of the jobs in ascending order and then executing it. A combination of sorting followed by computation of the median to determine an optimal dynamic time quantum has been used in [4] and [5]. Likewise different authors [6], [7], [8], [9], [10] have suggested different advents to beat the shortcomings of the conventional round robin scheduling algorithm that is presently being used.

3. PROPOSED ALGORITHM

```
For every process in the ready queue
do
{
  Count [Pi] = 1;

Old_Tq=Tq;

  If (Bt [Pi] < = Tq)
     Apply classical Round Robin

  Else If (Bt [Pi] > Tq)
  {
        If (Noc ! = Count [Pi])
              Apply classical Round Robin
        Else
        {
        If (Lot [Pi] < = [(2/5)*Tq]
              {
                    Tq = Tq + Lot [Pi];
                    Bt [Pi] = Bt [Pi] - Tq;
              }

              Else if (Lot [Pi] < =  Hm)
              {
                    Tq = Tq + k;
                    Bt [Pi] = Bt [Pi] - Tq;
                    Tq = old Tq;
              }
              Else
              Apply classical round robin
        }
  }
}
```

4. FORMULAE AND ABBREVIATIONS USED

Noc =integer (Bt /Tq)
Lot=Bt - [Noc*Tq]
N = Number of Processes
Tq = Time Quantum
Bt = Burst Time
Lot = Left Over Time
Wt = Wait Time
Tat = Turnaround Time
Noc =Number of cycles
Cs = Context Switches
Hm =Harmonic Mean
Ceil(x) = it is the smallest integer which is not less than x and always gives the upper bound of x.

There are many criteria's based on which the CPU scheduling algorithms can be categorized. Below are these:-
A. Turnaround time-it is the time required for the execution of a particular process in the ready queue
B. Response time-it is time when the request is submitted till the request till the first response is devised.
C. Waiting time-it is the sum of the total time spent by a process in the ready queue.
D. Context switch-when the CPU switches from one process to another, this shifting is known as the context switching.

Average Tat= Sum Of The finish Time Of All Processes / No. Of Processes

Average Wt= Tat [P_i]-BT [P_i] / No. Of Processes

Hm=harmonic mean=ceil (N/ [∑ {1/Lot[[P_i]])

5. PROPOSED APPROACH

Our proposed approach is to chiefly reduce the turnaround time, wait time, context switches and to increase the throughput. In our suggestion we have used the left over time as the comparison parameter with another constant called 'Hm' whose value has been calculated using the harmonic mean of the remainder of the quantity Bt / Tq.
Hm =harmonic mean=ceil (N/ [∑ {1/Lot[P_i] }])
Number of cycles (Noc) = Bt [P_i] / Tq
Left over time (Lot) = Bt- (Noc * Tq)
Initially the burst time of the process is compared with the time quantum, if it is less than or equal to the time quantum then the conventional round robin is employed on the process. And if the burst time is greater than the time quantum, we will proceed as follows:

1) If the current process is not in its last but one cycle, the conventional round robin is employed on the job
2) Otherwise we will check the following conditions:
    - The remainder or the left over time is compared with $2/5^{th}$ of the time quantum, if smaller than the left over time is added to the time quantum.
    - Else the left over time is compared with the harmonic mean, if smaller the harmonic mean value is added to the time quantum and the job is executed.
    - If the above stated conditions turn out to be false than the conventional round robin is applied on the process.

In all the cases the value of the time quantum changes only for the particular processes for which the above stated condition satisfies and remains same elsewhere.

6. EXPERIMENTAL ANALYSIS

Below are some of the examples in which are proposed approach of the modified round robin has been illustrated:

A. Example 1

Time Quantum=10ns

Table 1 Ready Queue Table for Example 1

Processes	Burst Time(ns)	Left over time(ns)	Number of cycles
P1	21	1	2
P2	12	2	1
P3	34	4	3
P4	27	7	2
P5	18	8	1

1) Classical Round Robin:

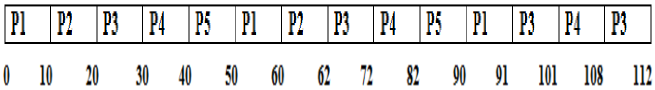

Figure 1 Gantt. Chart for the above ready queue of the processes

Average Tat= (91+62+112+108+90)/5

=92.6ns

Average Wt
= [(91-21) + (62-12) + (112-34) + (108-27) + (90-18)]/5 =70.2ns
Number of Cs=14

2) Modified Round Robin

Hm=harmonic mean=N / [∑ ({/Lot(P_i)}]
Where i =1 to N
N = Number of Processes
Tq =Time Quantum
P=Process
Noc =Number of Cycles
Count=Counter
Ceil Hm=5*[1/ {1+1/2+1/4+1/7+1/8}]
Ceil Hm=2/5*TQ=4ns

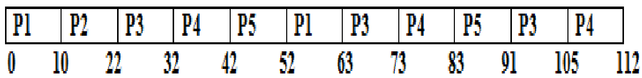

Figure 2 Gantt. Chart for the above ready queue of the processes

Average Tat= (63+22+105+112+91)/5
=78.6ns
Average Wt
= (63-21)+(22-12)+(105-34)+(112-27)+(91-18)/5
=56.8ns
Number of Cs=11

B. Example 2

Time Quantum=15ns

Table 2-Ready Queue Table for Example 2

Processes	Burst Time(ns)	Left over time (ns)	Number of cycles
P1	41	11	2
P2	23	8	1
P3	53	8	3
P4	21	6	1
P5	50	5	3
P6	34	4	2

1) Classical Round Robin

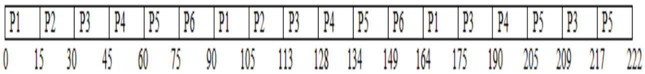

Figure 3 Gantt. Chart for the above ready queue of the processes

Average Tat= (175+113+217+205+222+163)/6
=182.66ns
Average Wt
=[(175-41)+(113-23)+(217-53)+(205-21)+(222-50)+(164-34)]/6
=145.33ns
Number of Cs=18

2) Modified Round Robin

Hm=harmonic mean=N / [∑ ({/Lot(P_i)}]

Where i=1 to N

N = Number of Processes

Tq =Time Quantum

P=Process

Noc =Number of Cycles

Count=Counter

Ceil Hm=6*[1/ {1/11+1/8+1/8+1/6+1/5+1/4}]
Ceil Hm=7
2/5*TQ=6ns

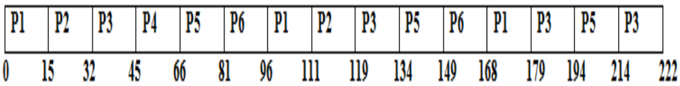

Figure 4 Gantt. Chart for the above ready queue of the processes

Average Tat = (179+119+222+66+214+168)/6
=161.33ns
Average Wt
=(179-41)+(119-23)+(222-53)+(66-21)+(214-50)+(168-34)/6
=124.33ns
Number of Cs=15

C. Example 3

Time Quantum=25ns

Table 3-Ready Queue Table for Example 3

Processes	Burst Time (ns)	Left over time (ns)	Number of cycles
P1	55	5	2
P2	62	12	2
P3	20	0	1
P4	18	0	1
P5	33	8	1
P6	27	2	1

1) Classical Round Robin

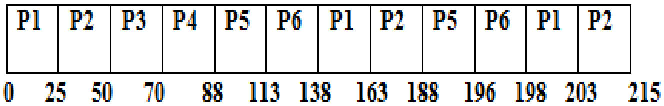

Figure 5 Gantt. Chart for the above ready queue of the processes

Average Tat = (203+215+70+88+196+198)/6
=161.66ns
Average Wt
=[(203-55)+(215-62)+(70-20)+(88-18)+(196-33)+(198-33)]/6
=125.83ns
Number of Cs=12

2) Modified Round Robin

Hm=harmonic mean=N/ [∑ ({/Lot(P_i)}]

Where i=1 to N

N = Number of Processes

Tq=Time Quantum

P=Process

Noc=Number of Cycles

Count=Counter

Ceil Hm=4*[1/ {1/5+1/12+1/8+1/2]

Ceil Hm=5

2/5*TQ=10ns

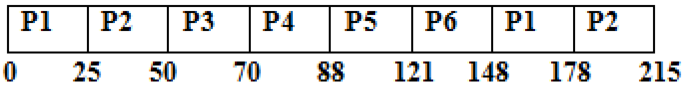

Figure 6 Gantt. Chart for the above ready queue of the processes

Average Tat= (178+215+70+88+121+148)/6
=136.66ns
Average Wt
=[(178-55)+(215-62)+(70-20)+(88-18)+(121-33)+(148-27)]/6
=100.5ns
Number of Cs=8

D. Example 4

Time Quantum=20ns

Table 4-Ready Queue Table for Example 4

Processes	Burst Time (ns)	Left over time (ns)	Number of cycles
P1	57	17	2
P2	73	13	3
P3	75	15	3
P4	31	11	1

1) Classical Round Robin

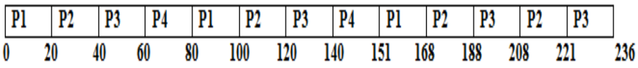

Figure 7 Gantt. Chart for the above ready queue of the processes

Average Tat= (168+221+236+151)/4
= 194ns
Average Wt= [(168-55) + (221-73)+(236-75)+(151-31)]/4
= 135.5ns

Number of Cs=13

2) Modified Round Robin

Hm =harmonic mean=N/ [∑ ({/Lot(P_i)}]
Where i=1 to N
N = Number of Processes
Tq =Time Quantum
P=Process
Noc=Number of Cycles
Count=Counter

Ceil Hm=4*[1/ {1/17+1/13+1/15+1/11}]
Ceil Hm=14

2/5*TQ=8

Gantt chart for the above ready queue of the processes:-

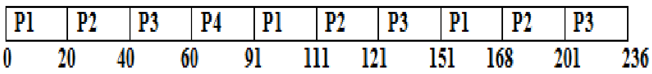

Figure 8 Gantt. Chart for the above ready queue of the processes

Average Tat= (168+201+236+91)/4
=174ns
Average Wt= [(168-55)+ (201-73)+(236-75)+(91-31)]/4
=115ns
Number of Cs=10

E. Example 5

Time Quantum=30ns

Table 5- Ready Queue Table for Example 5

Processes	Burst Time (ns)	Left over time (ns)	Number of cycles
P1	42	12	1
P2	67	7	2
P3	103	13	3
P4	60	0	2
P5	47	17	1

1) Classical Round Robin

Gantt chart for the above ready queue of the processes:-

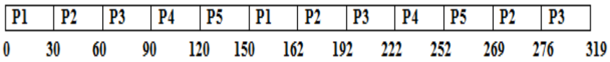

Figure 9 Gantt. Chart for the above ready queue of the processes

Average Tat= (162+276+319+252+269)/5

=255.6ns
Average Wt
= [(162-42)+(276-67)+(319-103)+(252-60)+(269-47)]/5
=189.8ns
Number of Cs=12

2) Modified Round Robin

Hm=harmonic mean=N/ [∑ ({/Lot(P_i)}]
Where i=1 to N
N = Number of Processes
Tq=Time Quantum
P=Process
Noc=Number of Cycles
Count=Counter
Ceil Hm=4*[1{1/12+1/7+1/13+1/17}]
Ceil Hm=12

2/5*TQ=12

Gantt chart for the above ready queue of the processes:-

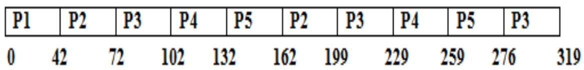

Figure 10 Gantt. Chart for the above ready queue of the processes

Average Tat=(42+199+319+259+276)/5
=219ns
Average Wt
= [(42-42)+(199-67)+(319-103)+(259-60)+(276-47)]/5
=151ns
Number of Cs=10

7. RESULT

Here we are presenting a tabular as well as a graphical form of the comparison between the classical RR and our approach for the examples given above.

A. Comparisons for Example 1

Table 6 Comparison Table for Example 1

Comparison Factor	Conventional Round Robin	Proposed Approach	Amount Of Refinement Observed
Average Turnaround time	92.6ns	78.6ns	14ns saved
Average Wait time	70.2ns	56.2ns	14ns saved
Context switches	13	10	3 context switches reduced

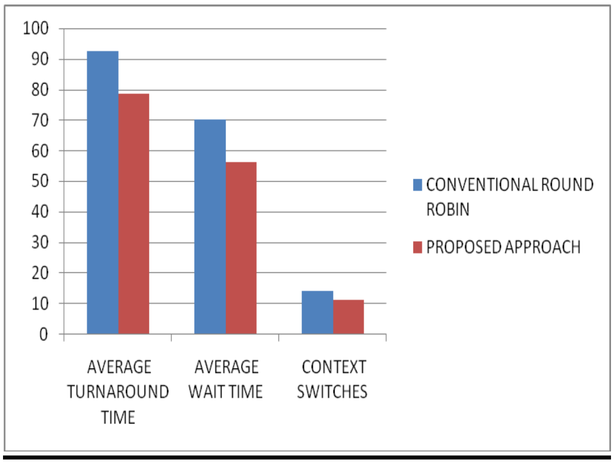

Figure 11 Comparison graph for Example 1

B. Comparisons for Example 2

Table 7 Comparison Table for Example 2

Comparison Factor	Conventional Round Robin	Proposed Approach	Amount Of Refinement Observed
Average Turnaround time	182.66ns	161.33ns	21.33ns saved
Average Wait time	145.33ns	124.33ns	21ns saved
Context switches	17	14	3 context switches reduced

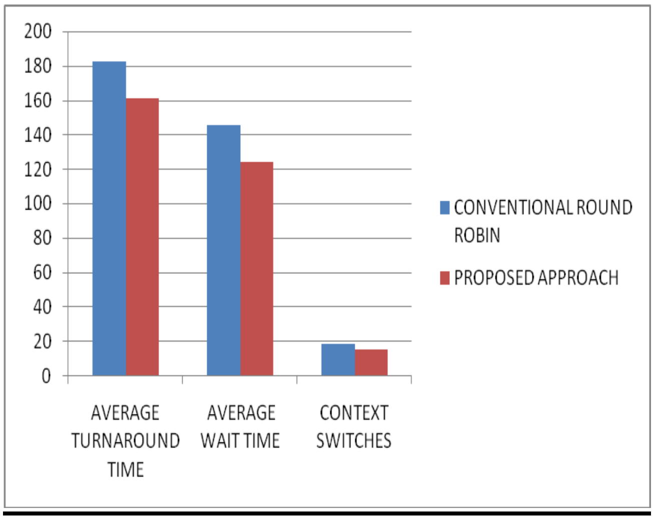

Figure 12 Comparison graph for Example 2

C. Comparisons for Example 3

Table 8 Comparison Table for Example 3

Comparison Factor	Conventional Round Robin	Proposed Approach	Amount Of Refinement Observed
Average Turnaround time	161.66ns	136.66ns	25ns saved
Average Wait time	125.83ns	100.5ns	25.33ns saved
Context switches	11	7	4 context switched saved

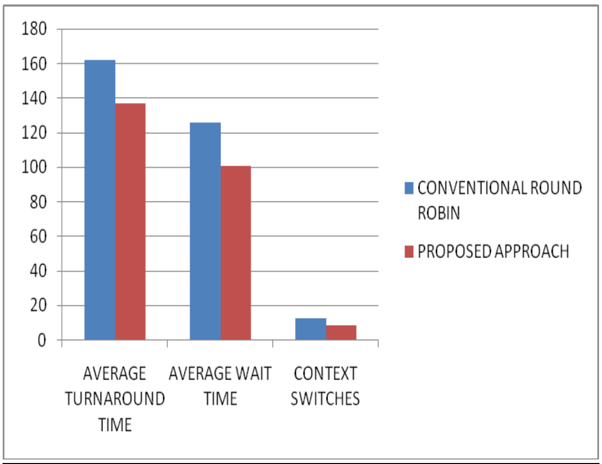

Figure 13 Comparison graph for Example 3

D. Comparisons for Example 4

Table 9 Comparison Table for Example 4

Comparison Factor	Conventional Round Robin	Proposed Approach	Amount Of Refinement Observed
Average Turnaround time	194ns	174ns	20ns saved
Average Wait time	135.5ns	115ns	20.5ns
Context switches	12	9	3 context switches saved

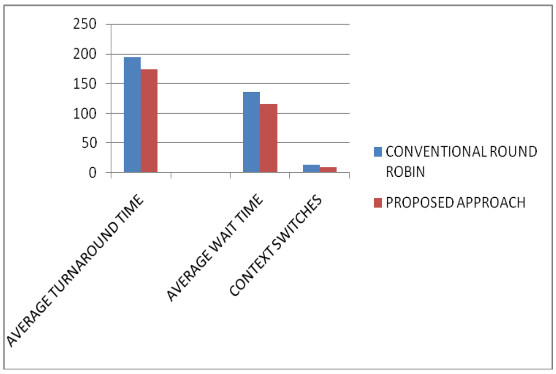

Figure 14 Comparison graph for Example 4

E. Comparisons for Example 5

Table 10 Comparison Table for Example 5

Comparison Factor	Conventional Round Robin	Proposed Approach	Amount Of Refinement Observed
Average Turnaround time	255.6ns	219ns	36.6ns saved
Average Wait time	189.8ns	151.8ns	38ns saved
Context switches	11	9	2 context switches reduced

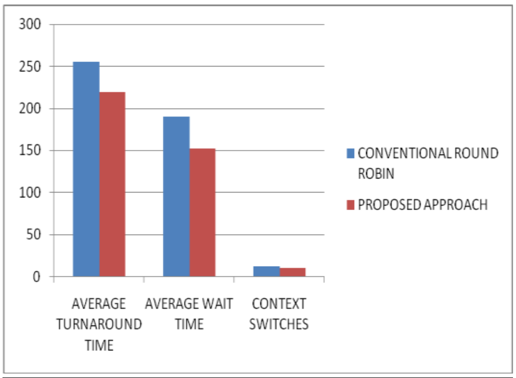

Figure 15 Comparison graph for Example 5

8. CONCLUSION

In this paper we have taken the round robin algorithm to improve its performance in terms of decreasing the quantities like the turnaround time, wait time and the number of context switches. We have supported our proposed approach using a number of examples which shows a clear edge of our approach over the conventional round robin algorithm. Since we are modifying the time quantum of only those processes which require only a fractional more time than the allotted time quantum cycle(s), the response time is also not affected and the overall performance is increased to a certain extent.

9. REFERENCES

[1] Ellen L. Hahne, Member, ZEEE, "An effective round robin algorithm using Min-Max dispersion measure", International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 01 January 2012.

[2] H.S Behera,Madhusmita Mishra,Simpi Patel," Weighted clustering based preemptive scheduling for real time system ",International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 05 May 2012.

[3] H. S. Behera, Rakesh Mohanty, Sabyasachi Sahu, Sourav Kumar Bhoi,"Comparative performance analysis of multi-dynamic time quantum round robin (mdtqrr)",Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2 No. 2 Apr-May 2011.

[4] Debashree Nayak Lecturer Gandhi Institute of Technology And Management,Bhubaneswar, Odisha, India, Sanjeev Kumar Malla Student Gandhi Institute of Technology And Management,Bhubaneswar, Odisha, India, Debashree Debadarshini Student Gandhi Institute of Technology And Management,Bhubaneswar, Odisha, India, "Improved round robin scheduling using dynamic time quantum", International Journal of Computer Applications (0975 – 8887),Volume 38– No.5, January 2012.

[5] H.S.Behera Professor Veer Surendra Sai University of Technology,Burla Sambalpur,IndiaR, Mohanty Professor Veer Surendra Sai University of Technology, Burla Sambalpur,India, Debashree Nayak Research Associate Veer Surendra Sai University of Technology, Burla Sambalpur,India, "A new proposed dynamic quantum with re-adjusted round robin scheduling algorithm and its performance analysis", International Journal of Computer Applications (0975 – 8887),Volume 5– No.5,August,2010.

[6] H.S.Behera Sr. Lecturer Veer Surendra Sai University Of Technology, Burla Sambalpur, India, Simpi Patel Student Veer Surendra Sai University Of Technology, Burla Sambalpur, India, Bijayalakshmi Panda Student Veer Surendra Sai University Of Technology, Burla Sambalpur, India, "A new dynamic round robin and srtn algorithm with variable original time slice and intelligent time slice for soft real time systems", International Journal of Computer Applications (0975 – 8887),Volume 16– No.1, February 2011.

[7] Shanmugam Arumugam and Shanthi Govindaswamy, Bannari Amman Institute of Technology and Science, Coimbatore, India: ECE Department, PSG College of Technology, Coimbatore, India, "Performance of the modified round robin scheduling algorithm for input-queued switches under self-similar traffic", The

International Arab journal of information Technology, Vol3,No. 2, April 2006.

[8] Yaashuwanth .C & R. Ramesh Department of Electrical and Electronics Engineering, Anna University Chennai, Chennai 600 025, "Intelligent time slice for round robin in real time operating systems", IJRRAS 2 (2) February 2010.

[9] Ji-Young Kwak, Ji-Seung Nam, and Doo-Hyun Kim," A Modified Dynamic Weighted Round robin cell scheduling algorithm", ETRI Journal, Volume 24, Number 5, October 2002.

[10] Ajit Singh, Priyanka Goyal, Sahil Batra," An optimized round robin scheduling algorithm for cpu scheduling", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 07, 2010, 2383-2385.

[11] Saroj Hiranwal, Computer Science and Engineering Suresh Gyan Vihar University Jaipur, Rajasthan, India, Dr. K.C. Roy roy.krishna@rediffmail.com ,Electronics and communication Engineering Pacific University Udaipur, Rajasthan, India "Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice", International Journal of Data Engineering (IJDE), Volume 2, Issue 3.

[12] H.S. Behera, Brajendra Kumar Swain, Anmol Kumar Parida, Gangadhar Sahu, "A New Proposed Round Robin with Highest Response Ratio Next (RRHRRN) Scheduling Algorithm for Soft Real Time Systems", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-1, Issue-3, February 2012.

[13] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash and Himanshu Sharma, College of Engineering and Technology, IFTM Campus, Lodhipur Rajput, Moradabad, UP, INDIA, "An Improved Round Robin Scheduling Algorithm for CPU scheduling", Rakesh Kumar Yadav et. al. / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1064-1066.

[14] B.Suresh, Prasad Reddy P.V.G.D, Software Technology Parks of India, Kakinada, India and C.Kalyana Chakravarthy, Department of CSE, M.V.G.R.College of Engineering, Vizianagaram, India, "Variable quantum deficit round robin scheduling for improved fairness in multihop networks" International Journal of Distributed and Parallel Systems (IJDPS) Vol.2, No.1, January 2011.

[15] Prof. Rakesh Mohanty, Prof. H. S. Behera, Khusbu Patwari, Monisha Dash, M. Lakshmi Prasanna, Department of Computer Science & Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur, Orissa, India," Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time Systems", (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 2, No.2, February 2011.

[16] Abraham Silberschatz, Peter Baer Galivin, Operating System Concepts, Fifth Edition.

[17] G.Henry,"The Fair Share Scheduler",AT&T Bell Laboratories Technical Journal,63(8),Oct.1984,pp.1845-1857

[18] T.Helmy and A.Dekdouk,"Burst Round Robin:As A Proportional-Share Sheduling Algorithm,"In Proceedings of The Forth IEEE-GCC Conference on Towards Techno-Industrial Innovations,pp.424-428,11-14 November 2007,at the Gulf International Convetion Center,Bahrain