Self-Regulated Priority based Round Robin Scheduling Algorithm

Md.Gulzar[1], Lt. K. Ravindra Babu[2], A.Vinaya Babu[3], Ph.D, S.Udaya Kumar[4], Ph.D
[1]PGStudent, Vagdevicollege of engineering,Warangal, [2]Associate Professor in CSE, KITS(S)
[3]Principal, JNTUCE, JNTUH, Kukatpally, Hyd, [4]Principal, GCET, Cheeryala, Keesara, Hyderabad, ,

ABSTRACT
The main aim of this paper is to present a new scheduling algorithm even thoughthere exists good scheduling algorithms. Each scheduling algorithm is having its own merits and demerits. The proposed algorithm overcomes the demerits of existing scheduling algorithms like high average waiting time, high average turnaround time, low throughput, high number of context switches. The proposed algorithm is a preemptive algorithm which takes a time quantum to execute the processes like round robin scheduling algorithm. But the time quantum is calculated automatically depending up on the average of all burst times. And to avoid the problem of starvation high priority should be assigned for short process.

General Terms
CPU Scheduling Algorithms, Round Robin Scheduling, Self-Regulated Priority based Round Robin Scheduling.

Keywords
Average waiting time, average turnaround time, context switches.

1. INTRODUCTION
CPU scheduling is the basic of multiprogrammed operating systems. The operating system switches the CPUamong processes to make the system more productive. To achieve thisCPU must be as busy as possible; the CPU time should not be wasted because it affects the throughput of the system. In general it canbe said that CPU must execute some process all the time [6].

To increase the performance the processes should be scheduled very efficiently. The scheduler deals with this. Whenever a CPUis idle it fetches a process from memory and the dispatcher gives the control over CPU to the selected process by scheduler. The scheduler selects the process from memory by using CPU scheduling algorithms [7].

1.1 Scheduling Criteria
Different CPU scheduling algorithms are exists, but each scheduling algorithm has properties; the choice of selecting a scheduling algorithm may favor one class of processor over another. For comparing scheduling algorithms many scheduling criteria have been suggested. The criteria are as follows [8-13]

1.1.1 CPU utilization
The CPU must be as busy as possible, it's utilization ranges from 0 to 100 percent.

1.1.2 Throughput
Throughput can be given as the number of processes that completes the execution per time unit.

1.1.3 Turnaround time
Turnaround time of a process can be given as the interval from the time of submission of a process to the time of completion.

1.1.4 Waiting time
The waiting time can be given as the time a process waits in the ready queue to acquire the processor.

1.1.5 Response time
Response time is the interval from the time of submission of the request to time of first response produced.

1.2Scheduling Objectives
It canbe concluded that a good scheduling algorithm for real time and time sharing system must possess following characteristics[1]

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

2. SCHEDULING ALGORITHMS

Scheduling Algorithms deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. There exist various processes scheduling algorithm like First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling Algorithm, Round Robin Scheduling Algorithm, Multilevel Queue Scheduling, and Multilevel Feedback Queue Scheduling [3].

- First come first serve scheduling algorithm is based on FIFO. And it is non-preemptive in nature.
- Shortest job first scheduling algorithm deals with executing shortest process first, it is both preemptive and non-preemptive scheduling.
- Priority scheduling is based on priority of a process where high priority process executes first, and it is preemptive scheduling.
- The Round robin scheduling algorithm is preemptive scheduling algorithm where every process executes up to a time quantum in FIFO order.
- In Multilevel queue scheduling the processes are permanently assigned to a queue on entry to the system.
- Multilevel feedback queue deals with those processes that move between queues.

The proposed algorithm for process scheduling is based on round robin scheduling. This will overcome the drawbacks of round robin scheduling algorithm. The name given to it is “Self-Regulated Priority Based Round Robin (SRPB RR) Scheduling Algorithm”

2.1 ROUND ROBIN SCHEDULING ALGORITHM

Round Robin scheduling algorithm is designed for time sharing systems. It is similar to FCFS scheduling but preemption is added to switch between processes. A small unit of time called as time quantum or time slice is defined. A time quantum is generally ranges from 10 to 100 milli seconds. The processesare stored in ready queue. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of 1 time quantum. If the process burst time is bigger than time quantum, then process execution will preempted and the process moves back at the end of ready queue and the scheduler selects the next process from ready queue and allocate the CPU to that process. This process continues till the ready queue becomes empty [4].

2.2.1Drawbacks

- If small time quantum is given the number of context switches will increase which affects the execution time of a process. If big time quantum is given it works in the same manner as first come first serve algorithm [2].
- The shortest burst time processes may have to wait for a long time which causes starvation.
- Can’t be used for real time applications which need faster execution [5].

3. PROPOSED ALGORITHM

In the proposed algorithm we consider at the beginning the ready queue is empty. All the processes enter into the ready queue at 0millisecond.
The proposed algorithm works in following steps

1. First we calculate the time quantum as average of all CPU burst time of processes in ready queue.

$$\text{Time Quantum (T.Q)} = \frac{\text{Sum of CPU burst times of process}}{\text{Number of processes in ready Queue}}$$

2. Sort the processes in ready queue in ascending order and assign high priority for short process.
3. Allocate the CPU to high priority process first and if its burst time is bigger than T.Q preempt the execution and give the CPU to next higher priority Process.
4. Repeat step 2&3 till ready Queue becomes empty.

3.1 Pseudo Code for Proposed Algorithm

Step 1: begin

Step 2: initialize q=0,count=1, awt=0, att=0

Step 3: read number of processes

Step 4: for each process i=1 to

4.1: read burst time

4.2: End for

Step 5: calculate time quantum

T.Q = avg of all process burst time

Step 6: sort the processes in ascending order of their burst time, give high priority for short process

Step 7: while count = 1

7.1: count=0;

7.2: for each process i=1 to n

7.2.1: if bt[i]>tq then go to 7.2.1.1 else go to7.2.2

7.2.1.1: bt[i]=bt[i]-tq;

7.2.1.2: increment cst and count=1;

7.2.2: if bt[i] !=0 then goto 7.2.2.1else go to 7

7.2.2.1: bt[i]=0;

7.2.2.2: increment cst and count=1;

7.3 end for and while

Step 8: for each process i=1 to n find waiting time, turnaround time, number of context switches, AWT and ATT.

Step 9: Print the result

Step10: Stop.

3.2 Flow Chart

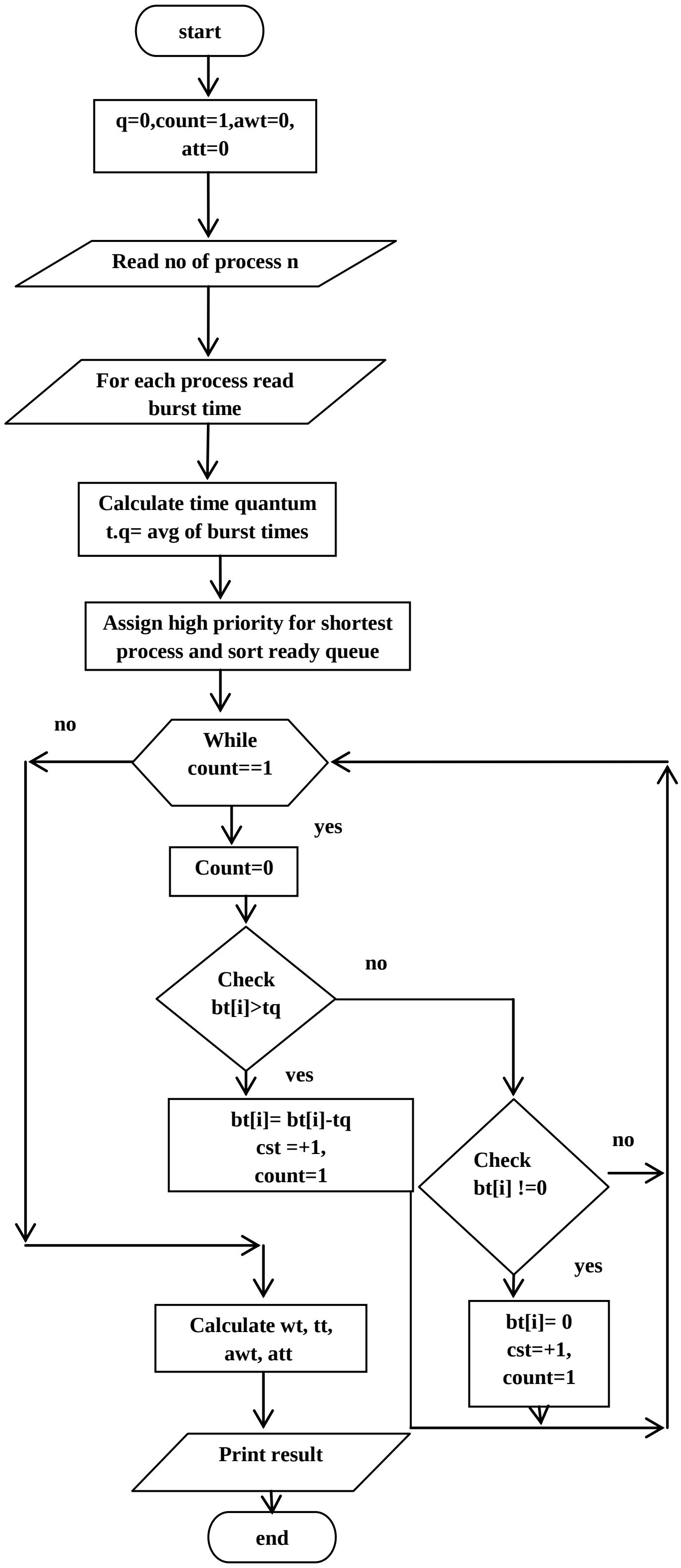

Fig 1: Flow chart for proposed algorithm

4. EXPERIMENTAL ANALYSIS

To evaluate the performance of proposed algorithm two examples are considered. These examples are solved with general round robin scheduling algorithm and with proposed self- regulated priority based round robin scheduling algorithm. And the analysis on these algorithms is done by comparing their results like average waiting time, average turnaround time and number of context switches.

4.1 Examplel1

Consider initially the ready queue is empty, and five processes arrived in ready queue at 0 ms, and their burst times are given as follows

PROCESS	BURST TIME
P1	15
P2	20
P3	5
P4	25
P5	10

4.1.1 Scheduling with round robin algorithm

In round robin scheduling the scheduler selects a process from ready queue in First in first out order (FIFO). And each process executes up to a time value of time quantum. Consider the time quantum is 10. To show the process scheduling Gantt chart [6] is used.

P1	P2	P3	P4	P5	P1	P2	P4	P4

0 10 20 25 35 45 50 60 70 75

Fig 2: Gantt chart for example1 with RR scheduling

From the above Gantt chart the results of round robin scheduling are as follows
The average waiting time is 36.00
The average turnaround time is 51.00
The number of context switches is 8

4.1.2 Scheduling with proposed algorithm

In the proposed algorithm the time quantum is the average of all burst times of the processes in the ready queue.
So the time quantum can be given as

$$T.Q = \frac{15+20+5+25+10}{5} = 15$$

And the processes in the ready queue will be sorted in ascending order, and high priority is assigned for shortest process. The higher priority or shortest burst time process executes first. The Gantt chat for this algorithm which shows process scheduling is as follows.

P3	P5	P1	P2	P4	P2	P4

0 5 15 30 45 60 65 75

Fig 3: Gantt chart for example1 with SRPB RR Scheduling

From the above Gantt chart the results of proposed algorithm is as follows

The average waiting time is 23. 00
The average turnaround time is 38.00
The number of context switches is 6.

Proposed algorithm is implemented in c programming language. When program is compiled with above input following output is generated.

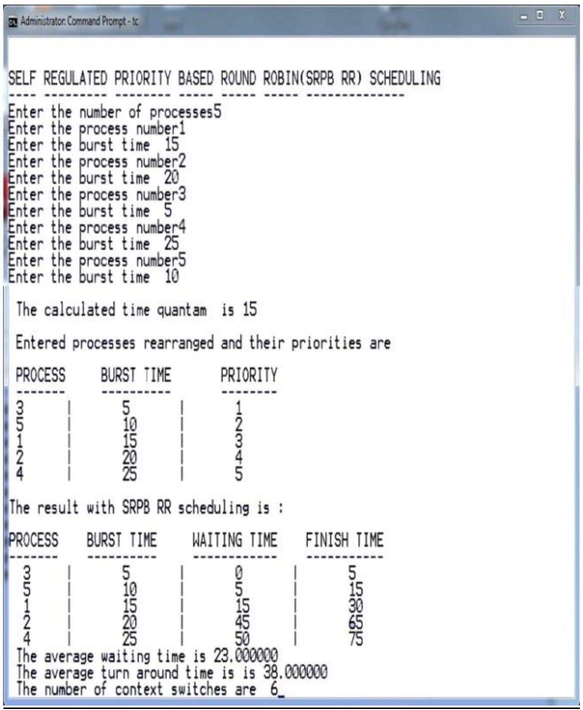

Fig 4: Result of example1 with SRPB RR scheduling Algorithm

4.1.3Analysis

Following table shows the analysis of results in above two algorithms

Table1. Analysis of two algorithms for example1

Algorithm	**Average waiting time (awt)**	**Average turnaround time (att)**	**Number of context switches**
RR	36.00	51.00	8
SRPB RR	23.00	38.00	6

Following chart as shown in figure5 graphically represents above analysis.

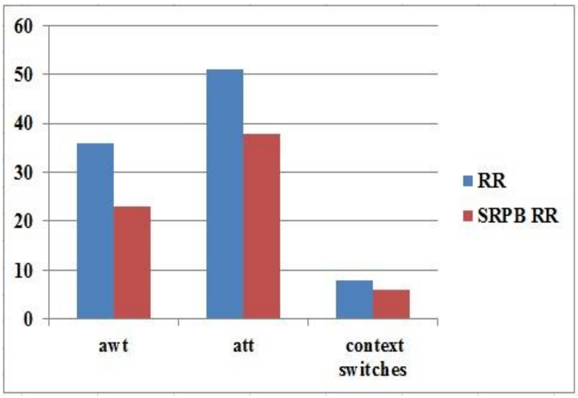

Fig 5: Analysis of example1 in two algorithms

4.2 Example2

Consider the ready queue is having four processes as follows.

PROCESS	BURST TIMES
P1	3
P28	
P3	5
P4	4

4.2.1 Scheduling with round robin algorithm

Consider the time quantum is 5. The following figure6 is the Gantt chart which shows the scheduling of above processes.

P1	P2	P3	P4	P2

0 3 8 13 1720

Fig 6: Gantt chart for example2 with RR scheduling

From the above Gantt chart the results of round robin algorithm is as follows

The average waiting time is 8.25.
The average turnaround time is 13.25.
The number of context switches is 4.

4.2.2 Scheduling with proposed algorithm

In proposed algorithm the time quantum for above processes is

T.Q =3+8+5+4 =5
4

4.2.3Analysis

Following table shows the analysis of results in above two algorithms.

The processes are sorted in ascending order of their burst times. Following figure7 is the Gantt chart which shows the scheduling of above processes with proposed algorithm.

P1	P4	P3	P2	P2

03 7 12 17 20

Fig 7: Gantt chart for example2 with SRPB RR scheduling

From the above Gantt chart the results of proposed algorithm is as follows

The average waiting time is 5.75
The average turnaround time is 10.50
The number of context switches is 4.

The generated output for above example is shown in figure8.

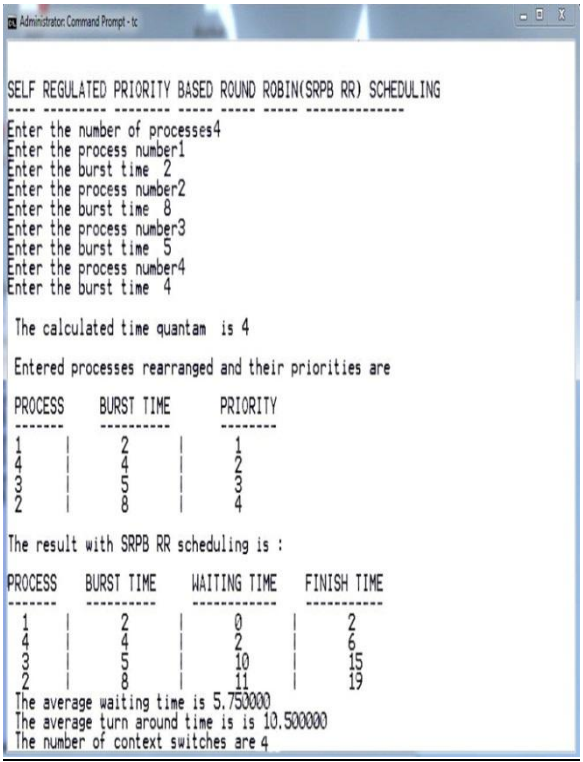

Fig 8: Result of example2 with SRPB RR scheduling Algorithm

Table2. Analysis of two algorithms for exampl2

Algorithm	Average waiting time (att)	Average turnaround time (awt)	Number of context switches
RR	8.25	13.25	4
SRPB RR	5.5	10.5	4

Following chart as shown in figure9 graphically represents above analysis.

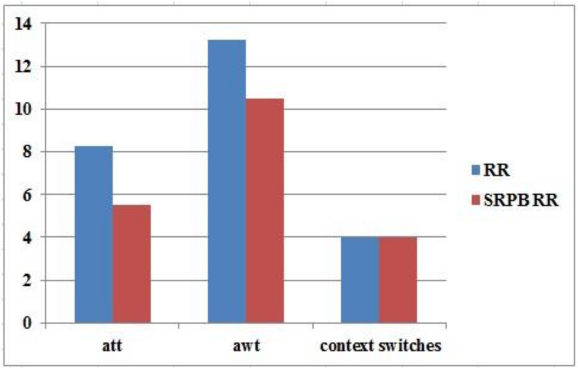

Fig 9: Analysis of example2 in two algorithms

From the above chart it can be seen that thereis same number of context switches, it is beacause the time quantum which is taken is same i.e 5 in both algorithms. But the average waiting time and average turnaround time is decreased with proposed algorithm.

5. CONCLUSION AND FUTURE WORK

With proposed algorithm the drawbacks of round robin scheduling are eliminated. It is shown by comparing the performance of two algorithms. And the problem of starvation is eliminated by allowing the shortest burst time process to execute first. The only limitation in proposed algorithm is the concideration that all processes will arrive in ready queue at once i.e at 0ms. It can be eliminated in future if proper work is done on this.

6. ACKNOWLEDGEMENTS

First author would like to thank other authors for their excellent guidance.

7. REFERENCES

[1] M.Ramakrishna. ”Efficient Round Robin CPU Scheduling Algorithm For Operating Systems”,International JournalOf Innovative Technology And Research (2320 –5547), Volume No. 1, Issue No.1,Page No.103-109, December-January 2013.

[2] Amit Kumar Sain. “Dynamical Modified R.R. CPU Scheduling Algorithm”, International Journal of Computer Trends and Technology(2231-2803), Volume No.4, Issue No.2, Page No.90-93,2013.

[3] Ankur Bhardwaj. “Comparative Study of Scheduling Algorithms in Operating System”, International Journal of Computers and Distributed Systems (2278-5183),Volume. No.3, Issue No.1,Page No.5-7 April-May 2013.

[4] Ajit Singh. “An Optimized Round Robin Scheduling Algorithm for CPU Scheduling”, International Journal on Computer Science and Engineering (0975-3397), Vol. 02, Paper No. 07, Page No. 2383-2385, May 2010.

[5] H.S.Behera. “A New Proposed Dynamic Quantum with Re-AdjustedRound Robin Scheduling Algorithm and Its PerformanceAnalysis”, International Journal of Computer Applications (0975 – 8887), Volume 5, Paper No.5, Page No.10-15, August2010.

[6] A.Silberschatz, P.B. Galvin and G. Gagne. “Operating System Principles”,John Wiley and Sons (978-81-265-0962-1), 7th Edition, India, Page No: 153, 154, 2008.

[7] Milan Milenkoviv. “Operating System Concepts and Design”,Tata McGraw hill (0-07-463272-8), 2nd edition,New Delhi., Page No-82, 1998.

[8] Ishwari Singh Rajput. “A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems” International Journal of Innovations in Engineering and Technology(2319 – 1058), Volume No. 1, Issue No. 3, October 2012.

[9] Abbas Noon. “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average”,International Journal of Computer Science Issues (1694-0814),Volume No. 8, Issue No.3, No. 1, Page No.224-229, May 2011.

[10] Vishnu Kumar Dhakad. “Performance Analysis Of Round Robin Scheduling Using Adaptive Approach Based On Smart Time Slice and Comparison With Srr”,International Journal of Advances in Engineering & Technology(2231-1963), Volume No.3, Issue No.2, Page No.333-339,May 2012.

[11] Abdulla Shaik. ”Shortest Time Quantum Scheduling Algorithm”,International Journal of Modern Engineering Research, (2249-6645), Volume No.2, Issue No.4, Page No.1548-1551,July-Aug 2012.

[12] Rami J.Matarne. ”Self-Adjustment Time Quantum in Round Robin Algorithm Depending onBurst Time of the Now Running Processes”, American Journal of Applied Sciences(1546-9239), Volume No. 6, Paper No. 10, Page No.1831-1837, June 2009.

[13] Sukumar Babu Bandarupalli.”A Novel CPU Scheduling Algorithm–Preemptive & Non-Preemptive”, International Journal of Modern Engineering Research (2249-6645), Volume No. 2, Issue No. 10, Page No. 4484-4490, Nov-Dec. 2012.