ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Modified Round Robin with improved Highest Response Ratio Next and Shortest Remaining Time Scheduling (MRHSS)

Neel Tushar Kapadia, Devanshi Anil Pooran Tahilramaney

Final Year Student, Dept. of Computer Science and Information Technology, VJTI, Mumbai, India

Final Year Student, Dept. of Computer Science and Information Technology, VJTI, Mumbai, India

ABSTRACT: The mechanism used to control the order in which jobs are to be executed by the CPU which affects CPU performance is called Scheduling. The design of scheduling algorithms is based on optimizing CPU utilization and throughput, and minimizing the turnaround time, response time, waiting time, and the number of context switches for a set of processes. In this paper, a variant of the Round Robin Algorithm with a new method of calculating quantum and that of selecting the process to be executed from the ready queue has been designed. Quantum is calculated using mean of burst times and, highest and lowest burst times of the processes. The process to be executed is decided based on the value of the response ratio of the processes in the ready queue at that instant of time. The response ratio is the ratio of sum of remaining time and the waiting time, to the service time of the process. This scheduling method gives a better result than other methods of improved round robin scheduling.

KEYWORDS: Operating System, Scheduling, Round Robin, Turnaround Time, Waiting Time, Context Switch, Highest Response Ratio Next, Shortest Remaining Time.

I. INTRODUCTION

OVERVIEW

An operating system (OS) is a system software that manages computer hardware and software resources and provides common services for computer programs. [6] An OS is used to provide an environment in which a user can execute programs in an efficient manner. The major functions of an OS include Process Scheduling, Memory Management and File Management. Other functionalities include Device Management, Protection and Security.

The fundamental function of an OS is scheduling. The system decides which task should be executed at a time using scheduling methods. When the CPU is idle, the next process to be executed is decided on the basis the method of Scheduling used. When a process cannot be executed at a given time, it is put into the ready queue and removed from the queue when it can occur. [4]

SCHEDULING CRITERIA [3]

The following criteria can be used to compare algorithms for scheduling:

Performance Related Criteria:

- Turnaround Time: The time interval between the arrival of the process and its completion. It includes execution time and the waiting time.
- Response Time: The time from the submission of the request until the response begins to be received. A scheduling discipline should attempt to achieve low response time.
- Deadlines: Deadlines can be set for process execution and the scheduling algorithm must try to meet all deadlines.

System Oriented Criteria:

- Throughput: The scheduling policy should attempt to maximize the number of processes completed per unit time. This is a measure of how much work is being performed. Although this depends on the average length of a process, it is also influenced by the scheduling policy which may affect utilization.
- Processor Utilization: This is the percentage of time that the processor is busy.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

- Fairness: In the absence of guidance from the user or system, processes should be treated the same and no process should suffer from starvation.
- Enforcing priorities: When processes are assigned priorities the scheduling algorithm should favour higher priority processes.
- Balancing resources: The scheduling policy should keep the resources of the system busy. Processes that underutilize stressed resources should be favoured.

A good scheduling algorithm should have:

- Lower average waiting time for processes.
- Lower average turnaround time (Tr).
- Lower ratio of Turnaround time by Service time (Tr/Ts).
- Maximum CPU Utilization.

TYPES OF SCHEDULING

Some of the scheduling algorithms referred to in this paper are:

- First-Come-First-Serve (FCFS)[3]: It is the simplest scheduling algorithm. The processes are executed in the order in which they arrive. The average waiting time for FCFS is quite high.
- Shortest Job First (SJF)[3]: The process from the ready queue that has the shortest burst time executes first. FCFS procedure is followed if 2 processes have the same burst time.
- Shortest Remaining Time (SRT)[3]: The process from the ready queue with the shortest remaining time first. FCFS procedure is followed if 2 procedures have the same burst time.
- Round Robin (RR)[3]: In this method a unit of time, called a quantum, is given to each process present in the ready queue. It maintains the fairness factor amongst the processes.
- Improved Round Robin (IRR)[1] : In this method, a fixed quantum is given to the processes. The processes execute in round robin manner. If at the end of the quantum, the currently running process has remaining time less than the quantum, the same process is executed further.
- Self-Adjusting Round Robin (SARR)[2] : In this method, the quantum is variable. The quantum is set as the median of the burst times of processes currently in the ready queue. At the end of the quantum, again the new quantum is the median of the burst times of the processes now in the ready queue and so on.

II. RELATED WORK

Related work done by the earlier authors in direction of the process scheduling and process management is described and explained here under.

In Year 2012, Manish Kumar Mishra[1] describes an improvement in RR. IRR picks the first job from the ready queue and allocate the CPU to it for a time interval of up to 1 time quantum. After completion of job time quantum, it checks the remaining CPU time of the currently running job. If the remaining CPU burst time of the currently running job is less than 1 time quantum, the CPU again allocated to the currently running process for remaining CPU time. In year 2009, Rami J. Matarneh[2] performed a work Self-Adjustment-Round-Robin (SARR) based on a new approach called dynamic time quantum, in this approach the time quantum repeatedly adjusted according to the burst time of the now-running processes. In year 2012, H.S. Behera[9] proposes A New proposed precedence based Round Robin with dynamic time quantum Scheduling algorithm for soft real time systems. In year 2012, H.S. Behera[10] proposed a new proposed round robin with highest response ratio next scheduling algorithm for soft real time system. In year 2013, Aashna Bisht[11] performed a work Enhanced Round Robin(ERR), in which modifying the time quantum of only those processes which require a slightly greater time than the allotted time quantum cycle. The remaining processes will be executed in the conventional Round robin manner. In year 2012, Ishwari Singh Rajput, Deepa Gupta[7] proposed priority based Round-robin CPU scheduling algorithms is based on the integration of round robin and priority scheduling. It retains the advantage of round robin in reducing starvation and also integrates the advantage of priority scheduling. The proposed new algorithm also implements the concepts of aging by assigning new priorities to the processes. In year 2012, P.Surendra Varma[8] performed a work, In which the improved version of SRBRR (Shortest Remaining Burst Round Robin) by assigning the processor to processes with shortest remaining burst in round robin manner using the best possible time quantum. In this paper the time quantum is computed with the help of median and highest burst time. In year 2010, Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash and Himanshu Sharma[12] performed a work. In this paper the new algorithm are used Round Robin with shortest Job first scheduling. In which allocate all processes to the CPU only one time as like present RR Scheduling algorithm. After second round select shortest job from the waiting queue and it shortest job assign to the CPU. After that next shortest job are selected.

III. PROPOSED ALGORITHM

In this algorithm, we have modified the approach of Round Robin algorithm to increase the throughput of the CPU. Based on observations, we devised a formula which will calculate quantum from the service times of the processes.
Quantum is calculated by taking the square root of the product of mean and the highest service time divided by lowest service time. For each time unit, processes which have arrived and are ready to execute enter the ready queue. This algorithm calculates the response ratio of each process in the ready queue. The response ratio is the ratio of sum of waiting time of the process and the remaining burst time to the total service time of the process. The process with the highest response ratio is selected for execution. In case 2 processes have the same response ratio, the one with the shortest remaining time is selected for execution.

IV. PSEUDO CODE

Step 1: All the process are sorted in ascending order of their arrival times.
Step 2: Calculate time quantum q.
m = Mean Service Time, h = Highest Burst Time, l = Lowest Burst Time
q = sqrt (m * h / l)
Step 3: While ready queue is NOT EMPTY
Calculate response ratio for all processes in ready queue
w = waiting time of process
r = remaining burst time
s = service time
R = (w + r)/s
Step 4: Select the process with maximum value of ratio for execution. If 2 processes have the same ratio, select the one with shortest remaining time
Step 5: Increment T(current time) and decrement r and if a process completes execution store the value of finish time f.

```
if r <= q
then
        T += r
        r = 0
        f = T
else
        r -= q
        T += q
```

Step 6: Go to step 3 till the ready queue is not empty.

V. EXPERIMENTAL ANALYSIS

The algorithm was executed on the following illustration and the results were contrasted with other algorithms.
A) All processes have arrival time 0
Case 1: Service Times in Increasing Order
Consider the set of processes A, B, C, D and E with arrival times 0 for all, and service times 5, 20, 30, 45,75.

MRHSS:

0 5 25 47 69 77 99 121 122 175

A	B	C	D	C	D	E	D	E

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0 5 35 45 65 85 95 115 135 140 175

A	B	C	D	E	C	D	E	D	E

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0 5 15 25 35 45 55 65 75 85 95 105 115 125 130 175

A	B	C	D	E	B	C	D	E	C	D	E	D	D	E

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = median of processes in ready queue)

0 5 25 55 85 115 130 175

A	B	C	D	E	D	E

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

Scheduling Algorithm	Average Turnaround Time	Average Waiting Time	Average Tr / Ts
MRHSS	80.8	45.8	1.96
RR	88	53	2.16
IRR	92	57	2.43
SARR	78	43	1.85

Case 2: Service Times in Decreasing Order

Consider the set of processes A, B, C, D and E with arrival times 0 for all, and service times 75, 45, 30, 20, 5.

MRHSS:

0 5 25 47 69 77 99 121 122 144 166 175

E	D	C	B	C	B	A	B	A	A	A

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0 20 40 60 80 85 105 125 135 155 160 175

A	B	C	D	E	A	B	C	A	B	A

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0 10 20 30 40 45 55 65 75 85 95 105 115 125 135 140 175

A	B	C	D	E	A	B	C	D	A	B	C	A	B	B	A

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = median = 30)

0 30 60 90 110 115 145 160 175

A	B	C	D	E	A	B	A

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

Scheduling Algorithm	Average Turnaround Time	Average Waiting Time	Average Tr / Ts
MRHSS	80.8	45.8	1.97
RR	127	92	6.28
IRR	112	77	4.5
SARR	130	95	7.48

Case 3: Service Times in Random Order

Consider the set of processes A, B, C, D and E with arrival times 0 for all, and service times 30, 75, 5, 20, 45.

MRHSS:

0	5	25	47	69	77	99	121	122	144	166	175
C	D	A	E	A	E	B	E	B	B	B	

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0	20	40	45	65	85	95	115	135	155	160	175
A	B	C	D	E	A	B	E	B	E	B	

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0	10	20	25	35	45	55	65	75	85	95	105	115	125	135	140	175
A	B	C	D	E	A	B	D	E	A	B	E	B	E	E	B	

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = 30)

0	30	60	65	85	115	145	160	175
A	B	C	D	E	B	E	B	

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

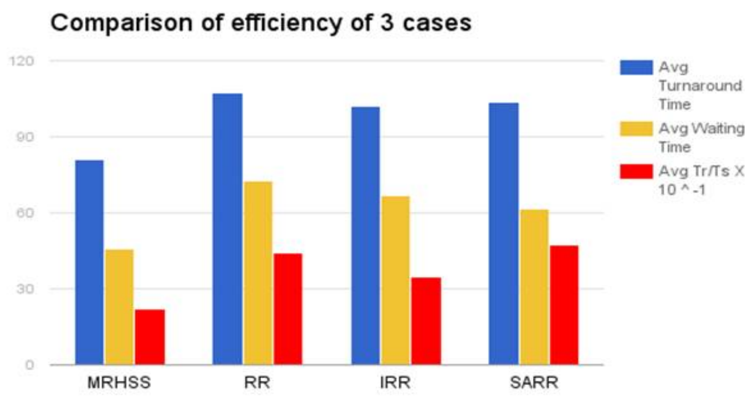
Scheduling Algorithm	Average Turnaround Time	Average Waiting Time	Average Tr / Ts
MRHSS	80.8	45.8	2.76
RR	108	73	4.26
IRR	102	67	3.47
SARR	103	47.2	4.83

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016



B) All process have varying arrival times

Case 1: Service Times in Increasing Order

Consider the set of processes A, B, C, D and E with arrival times 0, 3, 16, 32 and 40, and service times 5, 20, 30, 45 and 75.

MRHSS:

0 5 25 47 69 77 99 121 122 144 166 175

A	B	C	D	C	D	E	D	E	E	E

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0 5 35 45 65 85 95 115 135 140 175

A	B	C	D	E	C	D	E	D	E

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0 5 15 25 35 45 55 65 75 85 95 105 115 130 140 175

A	B	B	C	D	C	E	D	C	E	D	E	D	E	E

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = median of process in ready queue)

0 5 25 55 100 175

A	B	C	D	E

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

Scheduling Algorithm	**Average Turnaround Time**	**Average Waiting Time**	**Average Tr / Ts**
MRHSS	62.6	48.8	1.586
RR	69.8	34.8	1.78
IRR	65.8	30.8	1.67
SARR	53.8	19	1.34

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Case 2: Service Times in Decreasing Order

Consider the set of processes A, B, C, D and E with arrival times 0, 3, 16, 32 and 40, and service times 75, 45, 30, 20 and 5.

MRHSS:

0 22 44 66 71 91 99 121 122 144 166 175

A	B	C	E	D	C	B	B	A	A	A

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0 20 40 60 80 100 105 125 135 155 160 175

A	B	C	A	D	E	B	C	A	B	A

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0 10 20 30 40 50 60 70 75 85 95 105 115 125 135 140 175

A	B	A	C	B	A	D	E	C	B	A	D	C	B	B	A

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = median of processes in ready queue)

0 75 100 120 140 150 162 170 175

A	B	B	C	C	D	D	E

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

Scheduling Algorithm	**Average Turnaround Time**	**Average Waiting Time**	**Average Tr / Ts**
MRHSS	93.4	58.4	3.38
RR	116.8	75.8	5.23
IRR	107.8	72.8	4.03
SARR	119	82.8	8.17

Case 3: Service Times in Random Order

Consider the set of processes A, B, C, D and E with arrival times 0, 3, 16, 32 and 40, and service times 30, 75, 5, 20 and 45.

MRHSS:

0 22 27 49 57 77 99 121 143 144 166 175

A	C	B	A	D	E	B	E	E	B	B

The order of execution of processes by the modified algorithm is as shown above.

RR: (Quantum = 20)

0 20 40 45 55 75 95 115 135 155 160 175

A	B	C	A	D	E	B	E	B	E	B

The order of execution of processes by the traditional round robin algorithm is as shown above.

IRR: (Quantum = 10)

0 10 20 30 35 45 55 65 75 85 95 105 115 125 135 145 155 160 175

A	B	A	C	B	A	D	E	B	D	E	B	E	B	E	B	E	B

The order of execution of processes by the IRR algorithm is as shown above.

SARR: (Quantum = 30)

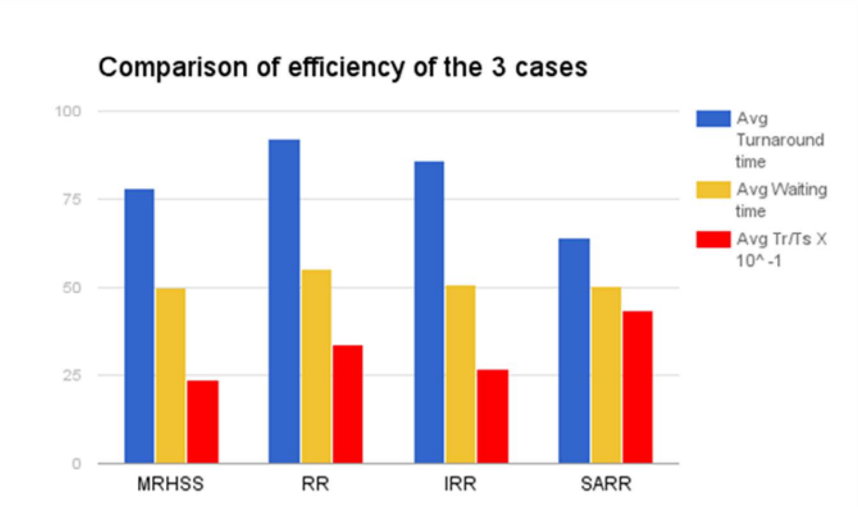
0 30 60 65 85 115 145 160 175

A	B	C	D	E	B	E	B

The order of execution of processes by the SARR algorithm is as shown above.

Comparison with other algorithms:

Scheduling Algorithm	**Average Turnaround Time**	**Average Waiting Time**	**Average Tr / Ts**
MRHSS	77.8	42.8	2.19
RR	89.8	54.8	3.11
IRR	83.8	48.8	2.32
SARR	72.8	49.8	3.52



VI. CONCLUSION AND FUTURE WORK

The comparison of efficiencies show that this algorithm is more efficient than other methods. We have covered many possible variations for arrival times of a given set of processes and we see that our algorithm is more efficient in most cases. The algorithm is starvation free since it implements a round robin approach.
In the future, it could be used as an alternative to the currently used algorithms.

REFERENCES

1. Manish Kumar Mishra, "Improved Round Robin CPU Scheduling Algorithm", *Journal of Global Research in computer science*, ISSN - 2229-371X, vol. 3, No. 6, June 2012
2. Rami J Matarneh, "Self- Adjustment Time Quantum in Round Robin Algorithms Depending on Burst Time of the Now Running Processes", *American Journal of Applied Sciences*, ISSN 1546-92396, (10):1831-1837, 2009
3. William Stalling, 2006, "Operating system", 5th Edition Pearson Education, ISBN : 81311703045.
4. "A.S. Tanenbaum, 2008" Modern Operating Systems, Third Edition, Printice Hall, ISBN: 13:9780136006633, PP: 1104.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

5. Silberschatz, A., P.B. Galvin and G.Gange, 2004,“ Operating System Concepts”. 7th Edn, John Wiley and Sons, USA, ISBN: 13:978-0471694663, PP: 944.
6. Wikipedia : https://en.wikipedia.org/wiki/Operating_system
7. Ishwari Singh Rajput,Deepa Gupta, “A Priority based round robin CPU Scheduling Algorithms for real time systems”, *International journal of Innovations in Engineering and Technology,* ISSN:2319-1058, Vol. 1 ISSUE 3 ,Oct 2012.
8. P.Surendra Varma, “A Best possible Time quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) algorithms”, *International Journal of advanced Research in computer science and software Engineering,* ISSN: 2277 128X, Vol. 2, ISSUE 11, November 2012.
9. H.S. Behera & Brajendra Kumar Swain, “A New proposed precedence based Round Robin with dynamic time quantum Scheduling algorithm for soft real time systems”, *International Journal of advanced Research in computer science and software Engineering,* ISSN:2277- 128X, Vol. 2, ISSUE 6, June 2012.
10. Brajendra Kumar Swain,H.S Behera and Anmol Kumar Parida, “A new proposed round robin with highest response ratio next scheduling algorithm for soft real time system”, *International Journal of Engineering and Advanced Technology,* ISSN: 2249-8958, Vol. - 1,ISSUE-3, Feb 2012.
11. Aashna Bisht, “ Enhanced Round Robin Algorithm for process scheduling using varying quantum precision”,*IRAJ International Conference-proceedings of ICRIEST- AICEE MCS*,29 th Dec 2013,pune India. ISBN: 978-93-82702-50-4.
12. Rakesh kumar Yadav, Abhishek K Mishra ,Navin Prakash and Himanshu Sharma, “ An Improved round robin scheduling algorithm for CPU scheduling”, *International Journal of computer science and Engineering,* ISSN: 0975-3397, Vol. 02, No. 04, 2010 , 1064-1066.

BIOGRAPHY

Neel Kapadia and Devanshi Tahilramaney are currently pursuing B.Tech in Information Technology at VJTI, Mumbai, India. Their expected graduation is in July 2017. Their research interests include Operating Systems and Data Structures and Algorithms.