Design and Performance Evaluation of Smart Job First Dynamic Round Robin (SJFDRR) Scheduling Algorithm with Individual Time Quantum for Each Process

Amit Kumar Gupta* Dr. Narendra Singh Yadav** Dr. Dinesh Goyal**
Resaerch Scholar* Associate Professor** Professor**
SGVU Jaipur* JECRC University Jaipur** SGVU Jaipur**

Abstract-Round Robin scheduling is mostly used CPU scheduling algorithm; it gives better result in comparison to other scheduling algorithm. But this algorithm may lead many problems directly related to time quantum. If selected time quantum is large, then the response time of the processes may turn in too high. On the other hand, if time quantum is short, it increases the number of context switch which may lead overhead of the CPU. In this paper, researcher proposed a new algorithm, called Smart Job First Dynamic Round Robin (SJFDRR) with individual time quantum for each process. The proposed algorithm calculates smart priority factor (SPF) and individual time quantum for each process. The factor SPF is calculated on the basis of user priority and system priority. The individual time quantum is calculated on the basis of burst time of the process. Based on the analysis, researcher has shown that the new proposed algorithm (SJFDRR) with individual time quantum solves the fixed time quantum problem and enhanced the performance of Round Robin.

Keywords: *Operating Systems, Multi Tasking, Scheduling Algorithm, Time Quantum, Round Robin.*

*****

I. INTRODUCTION

Today operating system are more complex, they are working on multitasking environment in which process are executed concurrently. So CPU scheduling is internal part of operating system design. When more than one process are ready to execute in ready queue then scheduler selects the process from the ready queue and dispatch it to CPU for execution. There are several classical scheduling algorithms available namely FCFS (First Come First Serve), SJF (Shortest Job First), Priority scheduling, Round Robin scheduling algorithm. These scheduling Algorithms are used to minimize the turnaround time, response time, waiting time and no of context switching. The operating system defines scheduling criteria, on the basis of these criteria the scheduling algorithm is analyzed and determines which scheduling algorithm is best.

II. SCHEDULING CRITERIA

There are many CPU scheduling algorithms having different properties, and the selection of a particular algorithm may depend upon on its performance matrices. In this section the researcher has described the scheduling criteria, on the basis of these criteria the performance of scheduling algorithm is evaluated [1, 2, 3, 4].

- **Context Switch**: A context switch occur when a process interrupt the normal execution sequence of another process. Then CPU stores all relevant information of interrupted process in Task Control Box (TCB). The context switch includes wastage of time, memory and scheduling overhead. So scheduling algorithm is designed in such way that it can minimize the number of context switches.
- **Throughput**: This term is defined as number of process completed per unit time. So scheduling algorithm is designed in such way that it can maximise the throughput.
- **CPU Utilization**: From the performance wise concern the CPU cannot be sit ideal. So scheduling algorithm is designed in such way that it cans maximum use of CPU as possible.
- **Turnaround Time**: It is the difference in the time of process when a process is ready to execute and when it complete its execution. So scheduling algorithm is designed in such way that it can minimize the turnaround time.
- **Waiting Time** it is the sum of all waiting done by a process in ready queue for execution. So scheduling algorithm is designed in such way that it can minimize the waiting time.
- **Response Time**: Response time is the time it takes to start its execution not the time it takes to output the response.

III. SCHEDULING ALGORITHM

There exist different Scheduling algorithms, each of them has advantages and disadvantages and as follows:

First-Come-First-Served (FCFS) FCFS is simple scheduling algorithm in which process are executed on the basis of their arrival time in ready queue. This scheduling algorithm is non preemptive in nature. The disadvantages of this algorithm is long waiting time, response time for high priority process.

Shortest-Job-First (SJF) In this algorithm the process which have minimum CPU burst time will schedule first. This algorithm can be implemented in two way on is preemptive and another one is non preemptive. This is also known as Shortest Remaining Time first (SRTF). This algorithm may lead a problem that we cannot predict how long a job will executed.

Priority Scheduling In This algorithm the process which has priority among the processes will schedule first. This algorithm may lead a problem of starvation which is defined as if high priority processes are regularly available in ready queue then waiting time for low priority may become infinite.

Round Robin (RR) algorithm which is the main concern of this research is one of the oldest, simplest, and most widely used scheduling algorithms. This algorithm works on time sharing phenomenon. A time slice is given to every process and every process will executed for particular defined time slice. New processes are added to at the last of ready queue. The scheduler pick the process from the starting point of the ready queue and set the timer to a defined time slice and also set an interrupt. If the process still not completed its complete execution within a time slice it will be preempted after a time slice and added at the end of ready queue. The round robin scheduling give the better response time, minimize waiting time and turnaround time, maximize throughput and CPU utilization [1, 2].

IV. LITERATURE SURVEY

A. Round Robin Scheduling

The simple RR scheduling algorithm is defined by following steps [1, 2]:-

1. The scheduler selects the processes from the ready queue in the manner of FCFS.
2. Every process will be allocated to the CPU for a fixed time quantum say k units.
3. If the process has not finished their execution within a time quantum then it will be added to the end of the ready queue.
4. Repeat the step 1 to 3 until the ready queue is empty.

Assume there are five processes P1, P2, P3, P4 and P5 with their burst time as shown in table I. Also assume that the round robin quantum is 5 ms.

TABLE I PROCESSES WITH CPU BURST TIME

Process Id	CPU Burst Time(ms)
P1	22
P2	18
P3	9
P4	10
P5	5

RR quantum=5

Gantt chart:

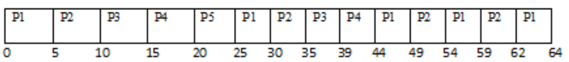

Figure 1 Gantt chart for Round Robin Scheduling

No of context switches=13

Average waiting Time=34 ms

Average turnaround Time=46.8ms

There are many researcher give many improved round robin algorithm. So the literature survey is presented

B. In 2009 Rami J.matarneh proposed a Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes [3] in which he has defined a technique for calculating optimal time quantum by calculating median for the set of processes in ready queue. He is also defined the minimum time quantum which is 25. It is described as if the calculated median is less than 25 then the time quantum will be 25 otherwise it will be equal to calculated median. This is done for minimize the number of context switches. The median is calculated by an equation defined below:

$$\bar{x} = \begin{cases} Y_{\frac{n+1}{2}} & \textit{if n is odd Eq (1)} \\ \frac{1}{2}\left(Y_{\frac{n}{2}} + Y_{\frac{n}{2}+1}\right) & \text{if n is even Eq (2)} \end{cases}$$

Where, Y is the number located in the middle of a group of numbers arranged in ascending order.

Then the optimal time quantum is calculated as

$$Q = \begin{cases} \bar{x}, & if\ \bar{x} \geq 25 \quad Eq\ (3) \\ 25, & if\ \bar{x} < 25 \quad Eq\ (4) \end{cases}$$

he has observed that his algorithm Self-Adjustment Time Quantum in Round Robin scheduling algorithm given better result in comparison to simple round robin scheduling (RR).

C. In 2011 Rakesh Mohanty, Manas Das, M.Lakshmi Prasanna ,Sudhashree proposed fittest job first dynamic round robin (FJFDRR) scheduling algorithm [5]. In this paper Rakesh Mohanty has calculated a factor f for every process. The process which has smallest f value will be schedule first. The factor f is calculated on the basis of given user priority (UP), system priority (SP) which is assigned as lowest burst time has highest priority, user weight (UW which is chosen as 60%) and system weight (SW which is chosen as 40%). The factor f is calculated as

f = UP * UW + SP * BW ------------------------Eq(5)

The dynamic time quantum is calculated as

TQ = median (remaining burst time of all the processes

He has calculated that his algorithm fittest job first dynamic round robin (FJFDRR) scheduling algorithm given better result in comparison to priority based static round robin scheduling PBSSR.

D. In 2012 Manish Kumar Mishra, Abdul Kadir Khan proposed An Improved Round Robin CPU Scheduling Algorithm (IRR) [15]. In this paper Manish Kumar Mishra propose an improved round robin scheduling algorithm (IRR) in which scheduler select the process from ready queue and allocate CPU for 1 time quantum. If the remaining burst time of currently running process is less than 1 time quantum then allocate CPU again for currently running process for remaining burst time otherwise the currently running process add at the end of ready queue.

They have observed that their algorithm improved round robin scheduling algorithm (IRR) scheduling algorithm given better result in comparison to simple round robin scheduling (RR).

E. In 2014 M. Ramakrishna, G.Pattabhi Rama Rao proposed Efficient Round Robin CPU Scheduling Algorithm for Operating Systems [18]. In this paper they have presented an efficient round robin CPU scheduling algorithm in which they have executed their algorithm in two steps. First they have arrange all the process in ascending order of the burst time and assign the priority to each process in the manner that lowest burst time process have assigned the highest priority and highest burst time assigned the lowest priority. Second after the completion of first cycle all the process are arranged in ascending order of their remaining burst time and assign the priority in the manner that lowest remaining burst time has assigned highest priority and highest remaining burst time assigned lowest priority.

They compared their algorithm efficient round robin CPU scheduling algorithm with simple round robin and found that their algorithm gives better result in comparison to RR scheduling.

F. In 2014 Manish Kumar Mishra, Dr. Faizur Rashid proposed An Improved Round Robin Scheduling Algorithm with Varying Time Quantum [17]. In this paper they have arranged all the process in the ascending order their burst time. Schedule the process in Round Robin manner with assigned time quantum is equal to the first process in ready queue. After the completion of first cycle again processes are arranged in ascending order of their remaining burst time and allocate the time quantum is equal to the burst time of first process.

They compared their algorithm improved round robin CPU scheduling algorithm with varying time quantum with simple round robin and found that their algorithm gives better result in comparison to RR scheduling

G. In 2014 Dr. R. Rama Kishore, Arpana Saxena proposed An Efficient Multi Parametric CPU Scheduling Algorithm for Single Processor Systems [16]. In this approach first they have assigned a factor BT to each process as shortest burst time process assign highest value of BT and highest burst time process assign lowest value of BT. Second they have assigned a factor ATT to each process as shortest arrival time process assign highest value of ATT and highest arrival time process assign lowest value of ATT. Third they defined a precedence factor f which is calculated as

PF = (priority*0.8) + (Burst Time*0.7) + (arrival time * 0.2)---------------------------------------Eq(6)

Then all the processes are arranged in the ascending order of calculated precedence factor PF. The shorter PF value process will schedule first with the time quantum which is calculated as

$$\text{Time Quantum (TQ)} = n/(1/tq_1 + 1/tq_2 + 1/tq_3 + \dots + 1/tq_n \text{ ------ --------------------------------- Eq(7)}$$

Where tq_i is burst time of i^{th} process.

H. In 2014 Abdulzaraq Abdulrahim, Salisu Aliyu, Ahmad M Mustapha, Saleh E Abdullahi proposed An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm [19] in which scheduler select the process from ready queue and allocate CPU for 1 time quantum. If the remaining burst time of currently running process is less than or Equal to time quantum then allocate CPU again for currently running process for remaining burst time otherwise the currently running process add at the end of ready queue.

They have observed that their algorithm An Additional improved round robin scheduling algorithm (AAIRR) scheduling algorithm given better result in comparison to simple round robin scheduling (RR) and Improved Round Robin (IRR) Algorithm.

I. In 2015 Rahul Joshi, Sashi Bhushan Tyagi Proposed Smart Optimized Round Robin (SORR) CPU Scheduling Algorithm [20] in which they have arranged all the process in ascending order of their burst time. Then they have calculated the smart time quantum as follows

(a) First they have calculated mean of burst time of all the processes.
(b) Second the smart time quantum (STQ) is calculated as follows:
STQ = (Mean+ Highest burst time)/2

Then they have schedule the process with smart time quantum (STQ).

They have observed that their algorithm Smart Optimized Round Robin (SORR) CPU Scheduling Algorithm scheduling algorithm given better result in comparison to simple round robin scheduling (RR) , Improved Round Robin (IRR) Algorithm and an Additional Improved Round Robin (AAIRR) Algorithm.

V. THE PROPOSED PLAN OF WORK:

In proposed plan of work, **first** researcher will calculate a smart priority factor ‘SPF’ is for every process. The process which has smallest ‘SPF’ value will be scheduled first. In this work every process has two types of priority one is user priority which is given by user itself (PRU) and second is the system priority which is defined by scheduling system in such a way that lowest burst time has highest system priority (PRS). The two important factors are also taken for calculating smart priority factor (SPF) which is user priority ratio (UPR) and system priority ratio (SPR). The user priority has more importance so the user priority ratio is given 55% weight and system priority ratio is given 45% weight. Suppose that all the processes has arrived at same

ISSN: 2321-8169
63 – 68

time i.e. arrival time = 0. Then Smart Priority Factor 'SPF' is calculated by following equation.

SPF=PRU*UPR+PRS*SPR--------------------- Eq(8)

Second researcher has defined Process time quantum (PTQ) for each process which is calculated by following steps.

Step1: Define Process burst weight (PBW) for each process in such a way that the smallest burst time process will be given more weight and initially remaining burst time (RBT) of each process is equal to its burst time.

N is the number of processes which has remaining burst time >0 and initially it is equal to number of process.

Step2: Calculate Average Burst time (ABT) by following formula

ABT= (Sum of Remaining burst time of Processes)/N

Step3: If N > 2 then go to step4 and step 5 otherwise go to step6

Step4: Calculate Smart Time Quantum (STQ) for each process (which has remaining burst time >0) by following formula:

(STQ_i) = Ceiling $((ABT*PBW_i)/PRU_i)$

Where

PBW_i= Process burst weight of i^{th} process

PRU_i= User Priority of i^{th} process

Step5: Then assign the process Time quantum to each process by following rules

IF $(STQ_i \geq RBT_i)$

Then

$PTQ_i = RBT_i$

$RBT_i= 0$

N=N-1

ELSE

$PTQ_i = STQ_i$

$RBT_i = RBT_i - STQ_i$

Where RBT_i is remaining burst time of i^{th} process and go to step7 for next round time quantum calculation for the process which has remaining burst time > 0.

Step6: The Smart Time Quantum (STQ) is calculated by following equation for each process which has remaining burst time > 0.

STQ= ABT

IF $(STQ \geq RBT_i)$

Then

$PTQ_i = RBT_i$

$RBT_i= 0$

N=N-1

ELSE

$PTQ_i = STQ$

$RBT_i = RBT_i – STQ$

Then go to step7 for next round time quantum calculation for the process which has remaining burst time > 0.

Step7: If N>0 then go to step2 otherwise go to step8.

Step8: Calculate Average waiting time, Average turnaround time and no of context switch.

Case 1: We Assume five processes arriving at time = 0, with increasing burst time (P1 = 5, P2 = 12, P3 = 16, P4 = 21, p5= 23) as shown in Table-II. The Table-IV shows the output using FJFDRR algorithm and our new proposed algorithm (SJFDRR). Fig. 2 and Fig. 3 show Gantt chart for both the algorithms respectively

TABLE II PROCESSES WITH BURST TIME AND PRIORITY

Processes	Arrival Time	Burst Time	User Priority
P1	0	5	2
P2	0	12	3
P3	0	16	1
P4	0	21	4
P5	0	23	5

Gantt chart:

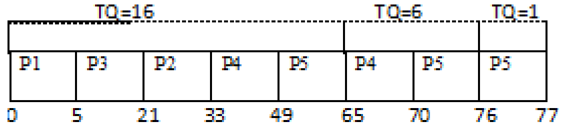

Figure-2 Gantt Chart Using FJDRR

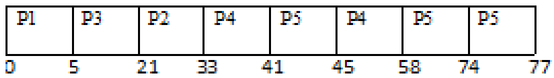

Figure-3 Gantt Chart Using SJFDRR

TABLE-III SHOWING PROCESS TIME QUANTUM IN EACH ROUND

Process	PTQ (I Round)	PTQ (II Round)	PTQ (III Round)
P1	5	0	0
P2	12	0	0
P3	16	0	0
P4	8	13	0
P5	4	16	3

TABLE-IV COMPARISION BETWEEN FJFDRR AND SJFDRR WITH INDIVIDUAL TIME SLOT FOR PROCESS

Algorithm	Avg.TAT	Avg.WT	CS
FJFDRR	25.8	41.2	7
SJFDRR with individual time slot for process	23.4	38.8	7

 ISSN: 2321-8169
63 – 68

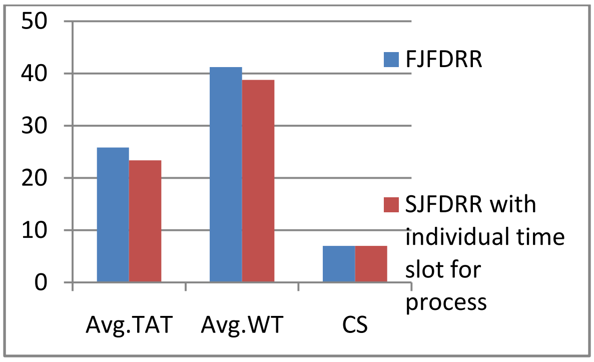

Figure 4 Comparison Graph between FJFDRR and SJFDRR with individual time quantum for each process

Case 2: We Assume five processes arriving at time = 0, with burst time (P1 = 30, P2 = 08, P3 = 24, P4 = 19, P5= 46) as shown in Table-V. The Table-VIII shows the output using FJFDRR algorithm and our new proposed algorithm (SJFDRR with individual time slot for each process Fig. 5 and Fig. 6 show Gantt chart for both the algorithms respectively

TABLE V PROCESSES WITH BURST TIME AND PRIORITY

Processes	Arrival Time	Burst Time	User Priority
P1	0	30	5
P2	0	8	3
P3	0	24	2
P4	0	19	1
P5	0	46	4

The Table is rearranged as increasing order of process burst time.

TABLE-VI PROCESS ARE ARRANGED IN ASCENDING ORDER OF BURST TIME

Processes	Arrival Time	Burst Time	User Priority
P2	0	8	3
P4	0	19	1
P3	0	24	2
P1	0	30	5
P5	0	46	4

Gantt chart:

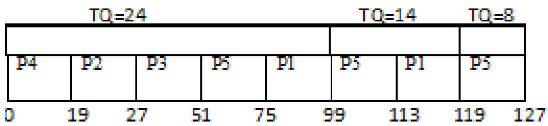

Figure-5 Gantt chart using FJFDRR

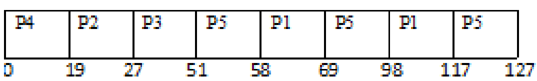

Figure-6 Gantt chart using SJFDRR

TABLE-VII SHOWING PROCESS TIME QUANTUM IN EACH ROUND

Process	PTQ (I Round)	PTQ (II Round)	PTQ (III Round)
P1	11	19	0
P2	8	0	0
P3	24	0	0
P4	19	0	0
P5	7	29	10

TABLE-VIII COMPARISION BETWEEN FJFDRR AND SJFDRR WITH INDIVIDUAL TIME QUANTUM FOR PROCESS

Algorithm	Avg.TAT	Avg.WT	CS
FJFDRR	43.2	68.6	7
SJFDRR with individual time slot for process	42.8	68.2	7

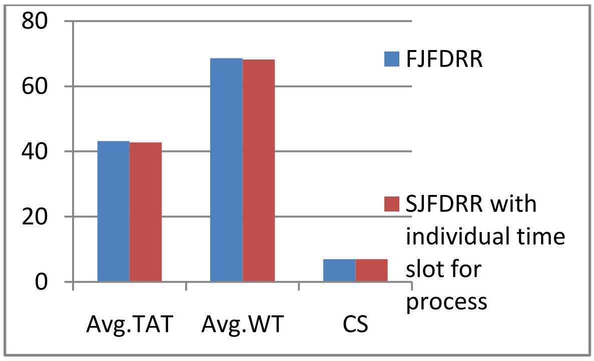

Figure 7 Comparison Graph between FJFDRR and SJFDRR with individual time quantum for each process

VI. CONCLUSION

From the analysis, it is founded that our proposed algorithm SJFDRR (smart Job First Dynamic Round Robin) with individual time slot performs better than the FJFDRR in terms of decreasing average waiting time and average turnaround time. But this work limited only for when arrival time of all process are zero. So the enhancement will in this algorithm **first**: research work for use different arrival time for the processes. **Second** if priority of process taking in to consideration than there will be a problem of **starvation.** So the research work will be focused on these two problems.

REFERENCES

[1] D.M. Dhamdhere operating Systems A Concept Based Approach,Second edition, Tata McGraw-Hill, 2006.

[2] Silberchatz, Galvin and Gagne, 2003. Operating systems concepts

[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.

[4] Ajit Singh, Priyanka Goyal, Sahil Batra “An Optimized Round Robin Scheduling Algorithm for CPU Scheduling” International Journal on Computer Science and Engineering Vol. 02, No. 07, 2010, 2383-2385

[5] Rakesh Mohanty, Manas Das, M.Lakshmi Prasanna ,Sudhashree “Design and performance evaluation of a new proposed fittest job first dynamic round robin (FJFDRR)scheduling algorithm”International Journal of Computer Information System Vol. 2 No. 2,2011

[6] Abbas Noon, Ali Kalakech, Seifedine Kadry “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011

[7] Rakesh Mohanty, H. S. Behera, Debashree Nayak, “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.

[8] Debashree Nayak,Sanjeev Kumar malla, Debashree Debadarshini “Improved Round Robin Scheduling using Dynamic Time Quantum” International Journal of computer Application (1975-8875) Volume 38-No. 5,January 2012.

[9] Rakesh Mohanty ,H.S. Behara,Khusbu Tiwari , Manas Das, M.Lakshmi Prasanna “Priority Based dynamic Round Robin(PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time System” International Journal of Advanced Computer Science and Application Vol. 2 No. 2, February ,2011.

[10] Rakesh Mohanty, H. S. Behera, Khusbu Patwari, Monisha Dash, “Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm”, In Proceedings of International Symposium on Computer Engineering & Technology (ISCET), Vol 17, 2010.

[11] H. S. Behera and et. al. 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 16(1):54–60, February 2011.

[12] H. S. Behera, Rakesh Mohanty, Sabyasachi Sahu, Sourav Kumar Bhoi. Design and Performance Evaluation of Multi Cyclic Round Robin(MCCR) Algorithm using Dynamic Time Quantum, Journal of Global Research in Computer Science, ISSN 2229-371X, Vol.2, No. 2, pp 48-53, February, 2011.

[13] H. S. Behera, Rakesh Mohanty, Jajnaseni Panda, Dipanwita Thakur Subasini Sahu, Experimental Analysis of A New Fair Share Wighted Time Slice(FSWTS) Scheduling Algorithm for Real Time Systems, Journal of Global Research in Computer Science, ISSN 2229-371X, Vol.2, No. 2, pp 54-60, February, 2011.

[14] Mohammed A. F. Al-Husain, Best-Job-first CPU Scheduling Algorithm, Information Technology Journal 6 (2):288-293, 2007. pp: 1104, 2008.

[15] Manish Kumar Mishra, Abdul Kadir Khan, “An Improved Round Robin CPU Scheduling Algorithm (IRR)” Journal of Global Research in Computer Science Vol. 6 No. 3,June 2012

[16] Dr. R. Rama Kishore, Arpana Saxena “An Efficient Multi Parametric CPU Scheduling Algorithm for Single Processor Systems” International Journal of Computer Science & Engineering Technology (IJCSET) Vol. 5 No. 5, May 2014.

[17] Manish Kumar Mishra, Dr. Faizur Rashid, “An Improved Round Robin Scheduling Algorithm with Varying Time Quantum” International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.4, No.4, August 2014.

[18] M. Ramakrishna, G.Pattabhi Rama Rao, “Efficient Round Robin CPU Scheduling Algorithm for Operating Systems”, International Journal of Innovative Technology And Research Volume No. 1, Issue No. 1, December-January 2013, 103-109.

[19] Abdulzaraq 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.

[20] Rahul Joshi, Sashi Bhushan Tyagi, “Smart Optimized Round Robin (SORR) CPU Scheduling Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7, July 2015.