OTSA: An Optimal Task Scheduling Approach

Reena Sharma[1], Antima Saxena[2]

[1]*M.Tech Scholar, Department of Computer Science & Engineering, Y.I.T., Jaipur,Rajasthan,India*
[2]*Asst. Prof., Department of Computer Science & Engineering, Y.I.T., Jaipur, Rajasthan,India*

Abstract – *Over the years, with rapid growth of digital devices and techniques the scheduling has attained focus of intensive research, since it directly influences the performance of a system. Typically, scheduling is an art of determining that which process has execute and access resources when there are multiple run-able processes are available in a system. This speculation is crucial for each sector in which there is a need to create a schedule by selecting, assigning or concatenating activities that are performed on some kind of resources. However, a lot of scheduling technique has been introduced over past decades but each one have its own unique limitation and still not a single scheduling algorithm is efficient for all kinds of applications. To fill this gap a new optimized scheduling approach has been introduced in this paper which improves resource utilization, throughput, scheduler efficiency at trim down rate of turnaround and waiting time.*

Key Words: Scheduler, Scheduling, Scheduling Algorithms, Context switching.

1. INTRODUCTION

Today, scheduling is a form of decision making that continuously investigate by researchers in various fields of management, industrial engineering, operations research and computer science. In recent years, scheduling research has had an increasing impact on practical problems, and a range of scheduling techniques have made their way into real-world application development. In computer science the term scheduling is one of the fundamental and challenging functions of an operating system design that make decision of giving resources between possible processes for maximize the performance of system as well as to minimize waiting and turnaround time [1]. As modern operating Systems are moving towards multitasking environments the term scheduling especially CPU scheduling becomes as an important issue in this field, requires careful attention to ensure fairness and avoid process starvation. In a computer system scheduler and dispatcher has allocates the process to CPU for a set time slice.

A scheduling scheme may be in a form of preemptive and non preemptive algorithms. As like name preemptive algorithms discontinue the execution of active process whenever a higher priority process has arrives in system ready queue. Once higher priority process complete its execution the interrupted process starts it execution again. On the other hand the other technique known as non preemptive scheme has executes process till its completion even a higher priority process arrives during its execution time [2]. Figure 1 has demonstrated the scheduling and process transition state in a computer system.

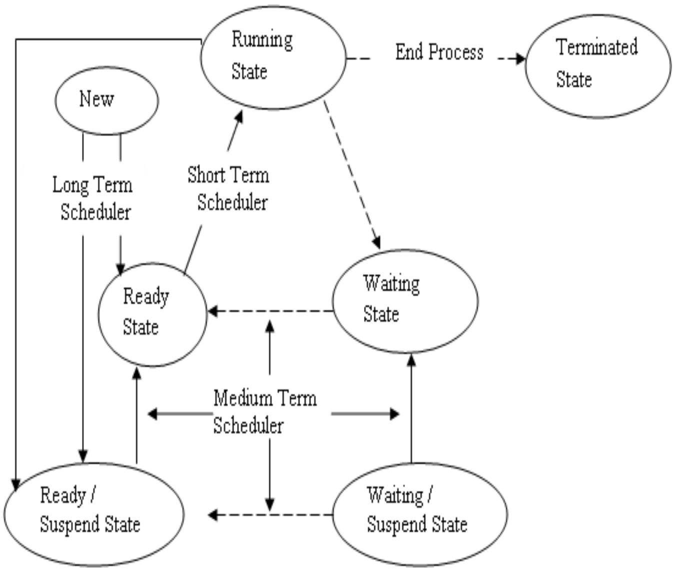

Figure 1 Scheduling and Process State Transition

Scheduling scheme affected by the following set of circumstances

1. When a process switches from the running state to the waiting state (for example, I/O request, or invocation of wait for the termination of one of the child processes)
2. When a process switches from the running state to the ready state (for example, when an interrupt occurs)
3. When a process switches from the waiting state to the ready state (for example, completion of I/O)
4. When a process terminates

Scheduling under 1 and 4 is non pre-emptive. On the other hand 2 and 3 are called pre-emptive. When CPU becomes idle the short-term scheduler (CPU scheduler) selects a process from ready queue for execution. The medium term scheduler determines when any process to be suspended or resumed. The job done by medium term scheduler is called swapping. Medium term scheduling is primarily deals with memory management; hence it is very often designed as a part of the memory management subsystem of an OS [3]. Instead of these two scheduling scheme one more scheduling term known as long term scheduler selects system admitted process for execution. It also decides on the ones which should exit. On the other hand this scheduler scheme supervises the degree of multiprogramming in multitasking systems. It follows certain policies through which decision is made which task will be selected if more than one task is submitted or whether the system can accept a new task submission. The compromise between degree of multiprogramming and throughput seems obvious by all processes for a fair share of over CPU as more the number of processes, lesser the time each of them may get on CPU for execution.

Different CPU scheduling algorithms has associates with different properties therefore each one algorithm carry out differently. The Criteria for a good scheduling algorithm depends on the following measures [4]:

1. **Maximum CPU Utilization:** Amount of time till CPU remains as busy as possible.
2. **Throughput:** Number of processes completed per unit time.
3. **Turnaround Time:** Total time that a process has taken in system for completion from its submission. In simple word turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.
4. **Waiting Time:** sum of the periods spent in the ready queue.
5. **Response time:** It is the time from the submission of the request until the first response is produced. It is the time it takes to start responding, not the time it takes to output the response.
6. **Fairness:** Equal CPU time for every process. Each and every process gets a fair share of the CPU.

The rest of this paper is organized as follows: Section 2 provides a snapshot of recent introduced scheduling algorithms. Section 3 demonstrates challenges and issues over accessible algorithm and the need of efficient scheduling scheme. Section 4 illustrated the proposed approach and section 5 presents simulation results analysis. Finally section 6 concludes the work of this paper.

2. ACCESSIBLE OPTIMIZED APPROACHES FOR CPU SCHEDULING: RELATED WORK

There are various researches going around the globe on enhancing the overall performance of CPU scheduling algorithms. However, each and every accessible scheduling methods has certain advantages and disadvantages and still there is no universal best CPU scheduling algorithm but a good number of researchers have proposed their unique contribution for improving such issues.

The authors of [5] have demonstrated a improved version of RR technique (IRR). In their designed approach they improved the QOS of traditional RR algorithm by modifying a process of execution. According to them the approach execute first process of ready queue for a set time quantum time and after completion if that process has requirement of low burst time form current set burst time then algorithm executes same procedure again without switching that process in tail of designed queue. In this case this process will finish execution and it will be removed from the ready queue. The scheduler then proceeds to the next process in the ready queue. Otherwise, if the remaining CPU burst time of the currently running process is longer than 1 time quantum, the process will be put at the tail of the ready queue. The CPU scheduler will then select the next process in the ready queue.

Another approach [6] has use FFGA (Fonseca and Fleming's Genetic Algorithm) with the aim to improve QOS of existing CPU scheduling algorithm. The authors of this work incorporated three parameters of CPU burst time; I/O devices service time, and priority of process instead of using one parameter of CPU burst time. The designed approach selects an execution process according to the system condition. To show the effectiveness of proposed approach they have compared performance with the traditional FCFS, RR, SJF and Priority techniques. For the comparison they use FCFS and RR technique with equal, prioritized way and for SJF and Priority algorithm they implement with pre-emptive and nonpreemptive fashion. The simulation results have demonstrated that proposed method has optimizes the average waiting time and response time for the processes.

A new preemptive CPU algorithm called SJRR [7] has introduced by different group of authors. The designed approach pre-empt the process on the base of their appearance time in ready queue. According to the authors of work their approach helps to improve the average waiting time of Round Robin algorithm in real time uni-processor-multi programming operating system. They simulates designed approach along with traditional FCFS, RR and SJF technique for illustrating the benefits of new designed algorithm.

For improving QOS of existing CPU scheduling mechanism the number of authors has presents different mathematical model for calculating [8-10]. They attempt to select an efficient quantum time slice for a process exe0ution and for calculating the waiting time and turnaround time. An Additional Improvement Round Robin (AAIRR) algorithm has proposed in [11]. The approach enhances performance of traditional RR technique by reducing the waiting and turnaround time of an executed process. The designed approach works in a similar way as classical RR technique but have some modification. It works in three stages: Stage 1: It picks the first process that arrives to the ready queue and allocates the CPU to it for a time interval of up to 1 time quantum. After completion of process's time quantum, it checks the remaining CPU burst time of the currently running process. If the remaining CPU burst time of the currently running process is less or equal to 1 time quantum, the CPU is again allocated to the currently running process for remaining CPU burst time. In this case this process will finish execution and it will be removed from the ready queue. The scheduler then proceeds to the next shortest process in the ready queue. Otherwise, if the remaining CPU burst time of the currently running process is longer than 1 time quantum, the process will be put at the tail of the ready queue. Stage 2: The CPU scheduler will then select the next shortest process in the ready queue, and do the process in stage 1. Stage 3: For the complete execution of all the processes, stage 1 and Stage 2 have to be repeated.

In [12], a new group of authors have presented a Smart and Optimized Round Robin CPU scheduling algorithm which improved on the An Advanced Improved Round Robin Scheduling algorithm [11] and Improved Round Robin CPU scheduling algorithm [5]. The simulation results illustrated that designed approach gives better results in terms of average waiting time, average turnaround time and number of context switches in all cases of process categories than the simple Round Robin CPU scheduling algorithm, Improved Round Robin CPU scheduling algorithm and the An Advanced Improved Round Robin CPU Scheduling algorithm. The authors of this investigation has incorporates a mechanism that set a process quantum time dynamically.

In the same context three different approached has been proposed for discover perfect time slice a new fuzzy approach has used in [13-15]. The authors of investigations have offered brand new approaches usage of most and minimal burst time of the set of strategies in the ready queue and calculating a modified time slice. A new variant of MLFQ algorithm has been presented in [16]. The approach assigned a set time slice to each queue for optimizing turnaround time. The authors have analyzed their proposed solution performance using dynamic time quantum and neural network over MLFQ using static time slice for each queue.

A new group of authors has made a New Multi Level Feedback Queue [NMLFQ] Scheduling approach [17]. To show the efficiency and effectiveness over the traditional algorithms the authors has simulated it with traditional Dependent Activity Scheduling Algorithm (DASA) and Locke's Best Effort Scheduling Algorithm (LBESA).

According to [18-20], DASA and LBESA are the decent, benefit accrual scheduling algorithms. Both of these algorithms are employed to utmost extent for the growth of mission critical systems. The better result of NMLFQ scheduler presents its superiority over the other used two algorithms.

A Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum (IRRVQ) has present in [21]. The presented scheduling algorithm coined enhancing CPU performance using the features of Shortest Job First and Round Robin scheduling with varying time quantum. For enhancing the QOS of traditional RR algorithm authors has focused on reducing the waiting and turnaround time of an process. They have done number of simulations to prove the efficiency of their proposed scheduling solution.

In [22], authors have presented a new scheduling solution known as EDRR (Efficient Dynamic Round Robin) algorithm. With the aim to enhance the performance of RR technique the approach has includes advantages of round robin CPU scheduling algorithm of less chance of starvation. Round robin CPU scheduling algorithm has high context switch rates, large response time, large waiting time, large turnaround time and less throughput, these disadvantages can be improved with new proposed CPU scheduling algorithm. A comparative simulation has done in between SRBRR (Shortest Remaining Burst Round Robin), ISRBRR (Improved Shortest Remaining Burst Round Robin) and new proposed EDRR CPU scheduling algorithm for presenting the effectiveness of proposed approach.

In same context for enhance the performance of existing RR technique a new scheduling algorithm OMDRRS presented in [23]. The approach mostly focus on reducing the number of context switching for better performance in terms of low waiting, turnaround and response time. The authors have analysis the performance of their designed approach with the help of ANOVA and t-test.

3. CHALLENGES AND ISSUES OF ACCESSIBLE SCHEDULING ALGORITHMS

However, a good number of approached has been presented for improving the performance of CPU scheduling and a lot of work are going in same direction but due to frequent demands and issues efficient scheduling is still as one of the challenges in the computer engineering field. The main problem scheduling algorithms is to determine a task from the task set to execute and also determining a processor which should be executed the task on it. Other issues include the following:

- Fail utilize the complete performance of CPU.
- Fruitless tests of schedulability
- Huge Overheads.
- Restricted task models for multiprocessor systems with a limited access permission policies for shared resources

4. PROPOSED OPTIMAL SCHEDULING APPROACH

The functional architecture of anticipated algorithm primarily confers more focus on the imperfection of classical RR technique in terms of minimize context switching, average waiting and turnaround time. Apart of this the intended approach trim down the cause of starvation by giving special consideration on the job execution priority, select an appropriate process for execution at dynamic time that reduce overheads of system. Initially the designed approach of this investigation contain a queue to lay up information of occurring process in a sorted order according to the required burst time of process, so that the process with least burst time will listed first. If two processes have equal burst times then they take place in queue on the base of their occurrence. The figure 2 has demonstrated the work steps of designed approach.

1. Check Process occurrence in system.
2. Insert occurred process in designed queue at it proper place, arrange according to process burst time.
3. Compute optimal CPU allotment time period for each process execution.
4. Select process from head position and allot CPU for execution.
5. If process complete its working before ending of set time period than select next process from designed queue for execution else compute current process remaining burst time.
6. If computed remaining burst time of current process is less than from set time period than execute same process again with a fresh set time period otherwise add it at the end position of designed queue and select the next process from head position.
7. Repeat step 1 to 6 until whole process not completed.

Figure 2 Working of Designed Approach

5. PERFORMANCE EVALUATION AND RESULT ANALYSIS

To evaluate the performance of designed approach over the traditional algorithms routines the designed approach has been simulated with two traditional algorithms FCFS and RR. Different parameters has taken into account for recital estimation which can be explained as

5.1 Turnaround & Average Turnaround Time

Typically, Turnaround time refers the total time that a process has taken in system from its submission to its complete execution. It includes the whole time that a process remains in system, total of waiting & execution time. However, it highly depends on the machine performance i.e. for a same process different machine may produce different turnaround time. The average turnaround time typically a median value of process which calculated by dividing the total turnaround time by process number. For evaluating performance of algorithms on the base of average turnaround time following formula has used.

$$ATT = \frac{(\sum_{i=1}^{NP} T_T)}{NP} \quad (1)$$

Where

ATT = Average Turnaround Time
NP = Number of Process in Queue.
T_T = Turnaround time of a process.

5.2 Waiting & Average Waiting Time

A waiting time is a period at which a process does not executes any activity in system, remains in ready queue to wait for its execution. It excludes time when process has executes or does I/O completion.

$$AWT = \frac{(\sum_{i=1}^{NP} W_T)}{NP} \quad (2)$$

Where

AWT = Average Waiting Time
NP = Number of Process in Queue.
WT = Waiting time of a process.

5.3 Result Analysis

To analyze the fair performance of designed approach over the other accessible scheduling algorithm a number of experiments have been carried out with different parameters. In each simulation each scheduling technique i.e. traditional FCFS, RR and designed approach has been evaluated with same parameters and outcomes has compared on the base of above discussed parameters. Apart to compare the performance of designed approach with classical FCFS and RR scheduling technique it has also compared with some of the current introduced scheduling algorithm results for demonstrating the effectiveness and efficiency of proposed algorithm.

For first experiment the following set of process with CPU-burst time in milliseconds has considered.

TABLE 1: Process with Burst Time

S.No.	Process	Burst Time (ms)
1.	P1	14
2.	P2	04
3.	P3	05
4.	P4	04
5.	P5	08

The obtain results, presented in figure 3 has clearly indicates the efficiency of proposed scheduling mechanism over the other classical techniques FCFS and RR.

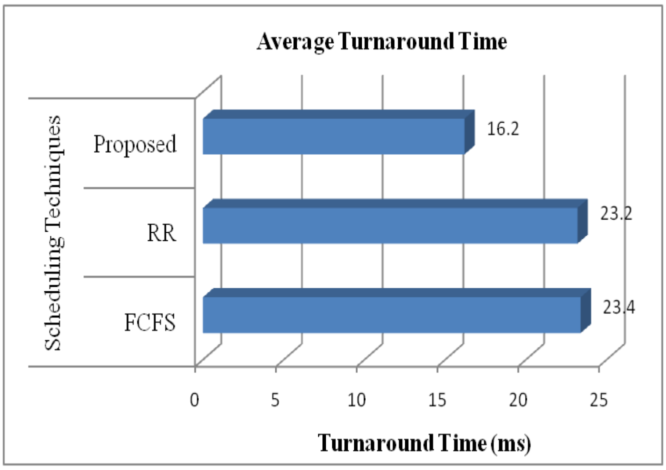

Figure3 Comparative Average Turnaround Time of FCFS, RR and Proposed Scheduling Scheme

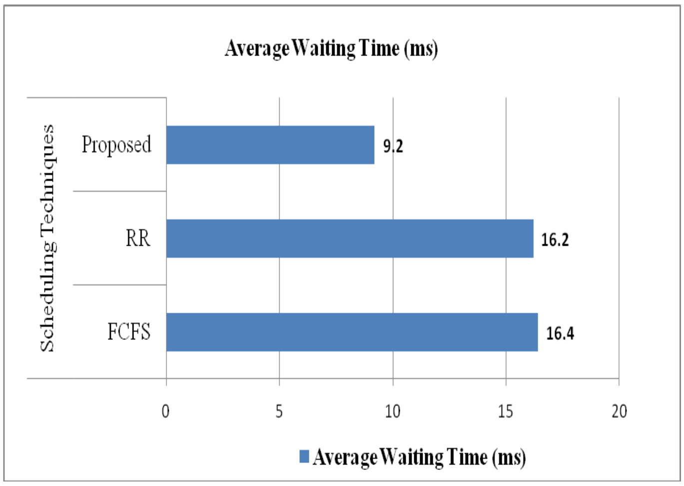

Figure 4 Comparative Average Waiting Times of FCFS, RR and Proposed Scheme

However, in literature of scheduling schemes a lot of investigators has illustrate that not a single algorithm is optimum solution for all type of jobs scheduling but from the above discussion it is clear that proposed approach is more suitable over the traditional algorithm First Come First Serve (FCFS) and Round Robin (RR).

To demonstrate the success and efficiency of designed approach over the other accessible solution a different simulations has also carried out. For fair comparison each simulation has done with the same process information as used by different authors in their investigations. The first comparative results has analyzed with the process information as used by Adaptive Round Robin (ARR) scheduling approach. The process description has been demonstrated in table 2.

TABLE 2 Process Information

S.No.	Process	Burst Time(ms)
1.	P1	14
2.	P2	45
3.	P3	36
4.	P4	25
5.	P5	77

The simulation results indicates that proposed approach has produced better results in term of average waiting and turnaround time over the ARR and traditional RR scheduling technique. Following comparative figures has demonstrates the comparison between evaluated schemes on the base of each process waiting time along with average waiting time (AWT).

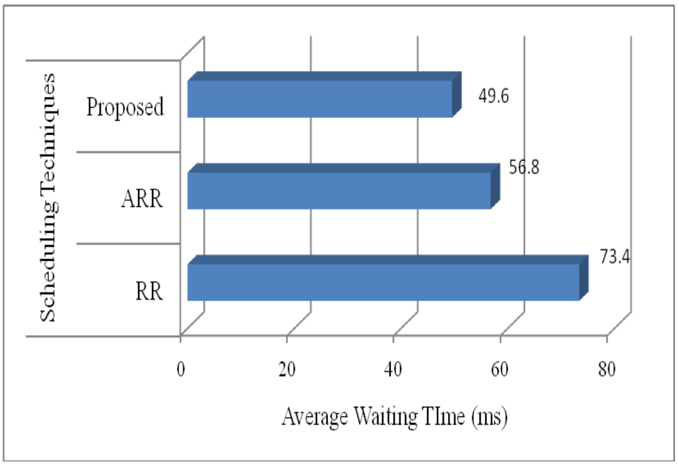

Figure 5 Comparative Average Waiting Time of Proposed Scheme over RR & ARR

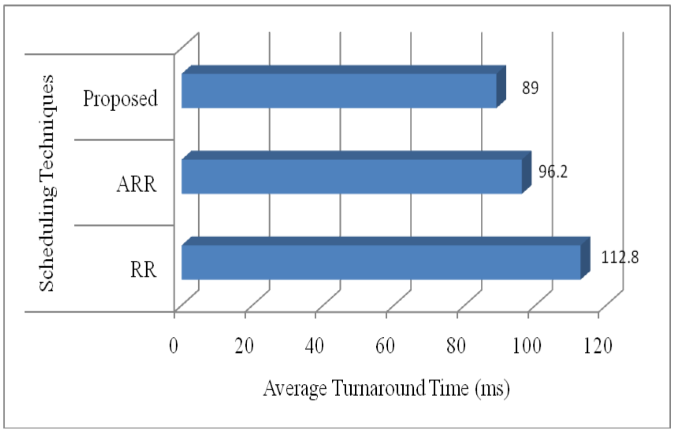

Figure 6 Relative Average Turnaround Time of Proposed Scheme over RR & ARR

6. CONCLUSION

This paper presents an overview of scheduling technique with its characteristics and associated issues. An efficient algorithm of scheduling guarantees for the objective of improving the system performance. However a lot of algorithms have been introduced over past few decades for trim down or improve the performance of system but most of the existing models of scheduling are fails in real time frame. To fill this gap a new endeavor has been made in this paper to improve the performance of scheduling. The designed scheduling approach has simulated and analysis with two most popular algorithm FCFS & RR. The results demonstrated that presented approach presents efficient outcomes over the traditional scheduling algorithms.

REFERENCES

[1] Sukumar Babu B., Neelima Priyanka N. & Dr. P. Suresh Varma "Optimized Round Robin CPU Scheduling Algorithm" Global Journal of Computer Science and Technology Interdisciplinary Volume 12 Issue 11 Version 1.0, pp. 21-26, 2012.

[2] Lingyun Yang, Jennifer M. Schopf and Ian Foster, ―Conservative Scheduling: Using predictive variance to improve scheduling decisions in Dynamic Environments‖, SuperComputing 2003, November 15-21, Phoenix, AZ, USA..

[3] Mohd Shoaib, Mohd Zeeshan Farooqui "A Comparative Review of CPU Scheduling Algorithms" Proceedings of National Conference on Recent Trends in Parallel Computing (RTPC - 2014).

[4] Sanjay Pandey, Indresh Gahoi, Neelesh Kumar Namdev" Fair Priority Round Robin with Dynamic Time Quantum: FPRRDQ" International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.3, May-June 2012 pp-876-881

[5] Manish kumar Mishra and Abdul kadir khan, "An Improved Round Robin CPU scheduling algorithm," Journal of Global Research in Computer Science Volume 3, No. 6, June 2012.

[6] Mehdi Neshat, Mehdi Sargolzaei, Adel Najaran, Ali Adeli "The New Method of Adaptive Cpu Scheduling Using Fonseca And Fleming's Genetic Algorithm" Journal Of Theoretical And Applied Information Technology 15th March 2012. Vol. 37 No.1

[7] Rakesh Patel , Mrs. Mili. Patel "SJRR CPU Scheduling Algorithm" International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 12, Dec.2013 Page No. 3396-3399

[8] Bashir Alam, "Fuzzy Round Robin CPU Scheduling Algorithm" , Journal of Computer Science, Vol. 9, No. 8, pp. 1079-1085, 2013.

[9] Lalit Kishor, Dinesh Goyal, "Time Quantum Based Improved Scheduling Algorithm" , Issue International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, 2013.

[10] P. Surendra Varma, "A Best possible time quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm" International Journal of Advanced Research in Computer Science and software Engineering, Vol. 2, No. 11, 2012.

[11] Abdulraza, 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 ISSN: 2277 128X.

[12] Rahul Joshi, Shashi 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

[13] G. Siva Nageswara Rao, N. Srinivasu, S.V.N. Srinivasu, G. Rama Koteswara Rao, "Dynamic Time Slice Calculation for Round Robin ProcessScheduling Using NOC" , International Journal of Electrical and Computer Engineering (IJECE) Vol. 5, No. 6, December 2015, pp. 1480~1485 ISSN: 2088-8708.

[14] G Siva, Nageswara Rao, et al., "Comparison of Round Robin CPU Scheduling Algorithm with Various Dynamic Time Quantum",International Journal of Applied Engineering Research, Vol. 9, No. 18, pp. 4905-4916, 2014.

[15] G Siva, Nageswara Rao, et al., "A New Proposed Dynamic Dual Processor Based Cpu Scheduling Algorithm", Journal of Theoretical and Applied Information Technology 20th March 2015, Vol.73, No. 2, 1992.

[16] Deepali Maste, Leena Ragha and Nilesh Marathe "Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling" International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 4 (2013), pp. 311-322

[17] M.V. Panduranga Rao , K.C. Shet "Assessment of Response Time for New Multi Level Feedback Queue Scheduler" International Journal of Computer Trends and Technology (IJCTT) – volume 13 number 3 – Jul 2014.

[18] Raymond Keith Clark, Jensen E. D and Rouquette N. F. "Software organization to facilitate dynamic processor scheduling", in IEEE Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Page(s): 122b, 2004.

[19] Michael B. Jones et al, "CPU Reservations and Time Constraints: Efficient, Predictable Scheduling of Independent Activities", in Proceedings of the sixteenth ACM Symposium on Operating Systems Principles, ACM SOSP '97, ISBN:0-89791-916-5, Page(s):: 198-211, 1997.

[20] Micro digital, "Simple Multitasking Executive", in proceedings of online document SMX RTOS, smxinfo, Embedded Software Outfitters, SMX , Symmetry Innovations, Australia, December 2004.

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

[22] Raman, Dr.Pardeep Kumar Mittal "An Efficient Dynamic Round Robin CPU Scheduling Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014

[23] Neetu Goel, Dr. R. B. Garg "Performance Analysis of CPU Scheduling Algorithms with Novel OMDRRS Algorithm" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 1, 2016