AN EFFECEINT MULTI PARAMETERIC CPU SCHEDULING ALGORITHM FOR SINGLE PROCESSOR SYSTEMS

Dr. R. RAMA KISHORE

USICT Department
Guru Gobind Singh Inderprastha University
New Delhi, India
E-mail: ram_kish@yahoo.com

ARPNA SAXENA*

MCA Department
AKGEC
Ghaziabad, India
E-mail: saxenaarpna@gmail.com

Abstract-- In today's world, computing systems serve many purposes and imagining a world without them is a nightmare. From the very beginning of the evolution of computer systems, their performance has been a subject of intense interest for computer programmers, engineers, buyers, managers and users. The measurement and optimization of computer system performance has consequently received a great deal of attention and analysis. So it gives the ability to do tasks at great speeds and it opens a new world of possibilities.

Central Processing Unit (CPU) plays a significant role in computer system by transferring its control among different processes. As CPU is a central component, hence it must be used efficiently. CPU Scheduling is a discipline which helps to gain deep insight into the complex set of policies and mechanisms used to govern the order in which tasks are executed by the processor. Many algorithms have been developed for the CPU scheduling of a modern multiprogramming operating system. This paper studies various existing and newly developed CPU scheduling algorithms. This research work involves the design and development of a new CPU scheduling algorithm (PSSDTQ). The primary objective of this algorithm is to optimize system performance according to the criteria demanded in the market.

Keywords--Process Scheduling, RR scheduling, average turnaround time, dynamic time quantum, average waiting time, Context Switch

I. INTRODUCTION

Operating System is the interface between the hardware and the user (application program). It controls and coordinates the use of the hardware among various application programs for various users [1]. Modern Operating Systems are moving towards multitasking environments which mainly depends on the efficiency of CPU scheduling algorithm since the CPU is the most effective part of the computer system. The CPU scheduling can be defined as the algorithm for determining which processes run on the CPU when there are multiple processes running in the system. In other words, it is the problem of deciding which computer process in the ready queue is to be allocated to the CPU for processing. Scheduling implies multiplexing a resource (i.e., CPU in this case) among several tasks to ensure all throughput requirements are met.

II. CPU SCHEDULING

The process scheduler is responsible for the assignment of CPU time to processes while making the system more responsive and having more throughputs etc.

The main aim of a scheduler is to keep the CPU busy as much as possible by executing a process until it must wait for an event, and then switch to another process[2].

Scheduling is a complex decision making activity because of conflicting goals, limited resources and the difficulty in accurately modeling real world scenarios [1, 2].

A. Types of CPU Schedulers

1) Long-term Scheduler

The long-term scheduler decides which jobs or processes are to be admitted to the ready queue. That is why it is also called job scheduler. Job scheduler selects processes from the queue and loads them into memory for execution. The primary goal of the job scheduler is to provide a balanced mix of processes, such as I/O bound and CPU bound. Thus, it controls the degree of multiprogramming (number of processes into a memory). On some systems, the long term scheduler may be absent or its minimal functionality is there.

2) Mid-term Scheduler

Medium-term scheduling involves suspending or resuming processes by swapping them out of or into memory. It temporarily removes process from main memory and place them on secondary memory (such as a disk drive) or vice versa. This is commonly referred to as “swapping out and swapping in" of processes in memory [1]. Since it removes the processes from the memory, therefore it reduces the degree of multiprogramming. The medium term scheduler is basically in-charge of handling the swapped out-processes. It can re-introduce the process into memory and execution can be continued.

3) Short-term Scheduler

It is also called CPU scheduler. It selects process among the processes that are present in the ready queue and allocates CPU to one of them. It makes scheduling decisions much more frequently than the long-term or mid-term schedulers.

This scheduler can be preemptive, or non preemptive. A scheduling discipline is non preemptive if, once a process has been given the CPU, the CPU cannot be taken away from that process [1].

A scheduling discipline is preemptive if, once a process has been given the CPU can taken away [1].

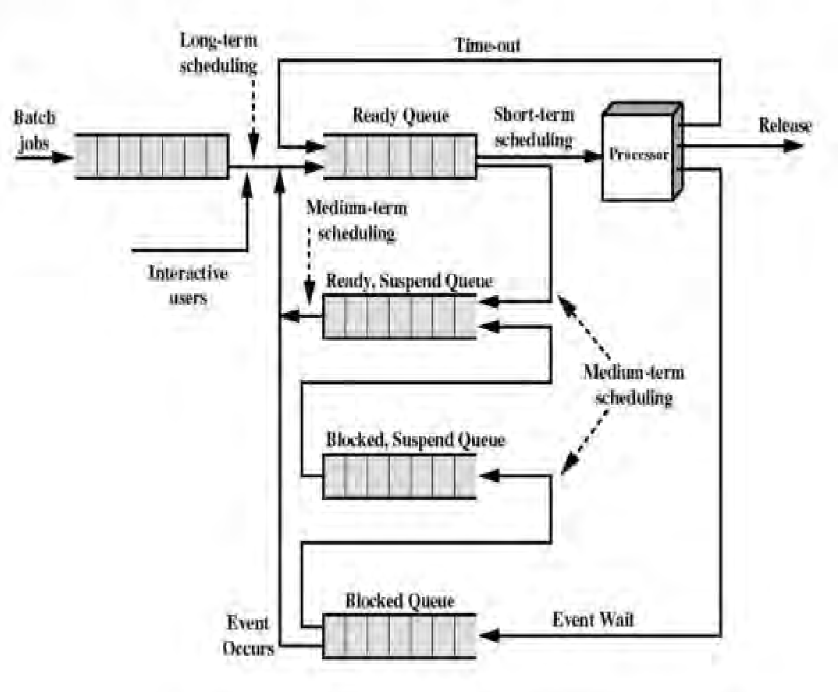

Figure 1: Various types of schedulers

B. Criteria for invoking a CPU Scheduling Algorithm

Process scheduling can be taken when:

1) Some process changes its state from running to waiting.

2) Some process terminates or exits from the system.

3) Some process changes its state from running to ready

4) Some process changes its state from waiting to ready

If only first two possibilities are used in an operating system, we called such an OS non-preemptive. Otherwise we say that OS is preemptive.

C. Criteria for evaluating a CPU scheduler

There are many different scheduling algorithms existing in present scenario which may vary in their efficiency according to the environment in which they are executing. This means what is considered as a good scheduling algorithm in some cases may not appear same for some other system, and vice versa. The Criteria for a good scheduling algorithm depends on the following measures [1]:

1) *CPU Utilization*: Scheduler should keep the system (or in particular CPU) busy cent percent of the time when possible. Usually, the goal is to maximize the CPU utilization.
2) *Throughput:* This is the number of processes completed per unit time. Usually, the goal is to maximize the throughput.
3) *Turnaround Time:* It is the mean time from submission to completion of a process. In other words it is amount of time a process takes from its creation till its termination from the system. A scheduler should minimize this time
4) *Waiting Time:* It is the amount of time spent by a process in a ready queue while waiting for the CPU. Usually, the goal is to minimize the waiting time.
5) *Response Time:* It is the amount of time elapsed when a request is submitted in the system and first response to the request starts coming. Usually, the goal is to minimize the response time.

D. CPU Scheduling Algorithms

Many basic algorithms for CPU scheduling have been designed. Some are discussed below [1, 2]:

1) *First-Come, First-Served (FCFS):* This algorithm allocates the CPU to the process that requests the CPU first. The processes are allocated to the CPU on the basis of their arrival time in the ready queue. It is easy to understand and implement, but it suffers from poor performance as average waiting time is high. It also suffers from convoy effect.
2) *Shortest-Job-First (SJF):* The SJF algorithm associates the length of the next CPU burst with each task. It then uses these lengths to schedule the task with the shortest time. The SJF algorithm may be implemented as either a preemptive or non-preemptive algorithms. Although experiments showed that it is the optimal algorithm but it is hard to implement because it demands the prediction of expected next CPU burst of processes. It may also leads to starvation of processes having large CPU burst time.
3) *Priority Scheduling (PS):* The PS algorithm associates with each process a priority and the CPU is allocated to the process based on these priorities. Usually, lower numbers are used to represent higher priorities. The process with the highest priority is allocated first. It may also be implemented as preemptive or non preemptive algorithm. It also suffers from starvation.
4) *Round Robin (RR):* The RR algorithm is designed especially for time-sharing systems and is similar to the FCFS algorithm. Here, a small unit of time (called time quantum or time slice) is defined. Each process is provided this quantum to execute. Once a process is executed for given time period that process is preempted and other process executes for given time period. Depending on the time quantum and the CPU burst requirement of each process, a process may need less than or more than a time quantum to execute on the CPU. In a situation where the process need more than a time quantum, the process runs for the full length of the time quantum and then it is preempted. The preempted process is then added to the tail of the ready queue.
5) *Multilevel Queues Scheduling:* In this case multiple queues are maintained for processes i.e., ready queue is partitioned in multiple queues. Each queue can have its own scheduling algorithms (like RR, priority, FCFS etc). Priorities are assigned to each queue. Scheduling must be done between the queues also.

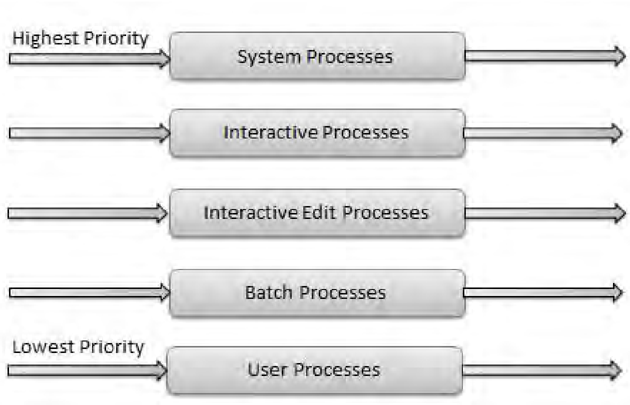

Figure 2[1]: Multilevel queue scheduling

6) *Multilevel Feedback Queues (MLFQ):* It is like multilevel queue scheduling algorithm except a feedback is there between the queues. Processes can move between queues as their priority changes. It can be used to give IO bound and interactive processes CPU priority over CPU bound processes. It can also prevent starvation by increasing the priority of processes that have been idle for a long time. This scheduler is defined by the following parameters:

a) number of queues

b) scheduling algorithms for each queue

c) method used to determine when to upgrade a process

d) method used to determine when to demote a process

e) method used to determine which queue a process will enter when that process needs service

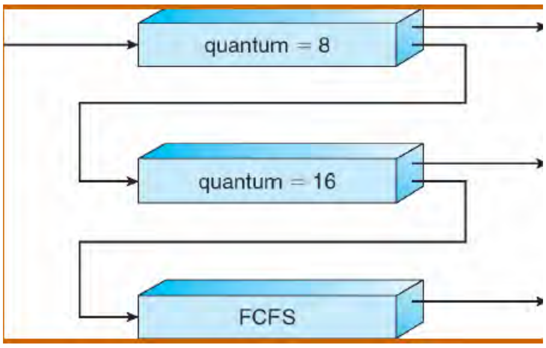

Figure 3[1]: MLFQ

But all these algorithms have one or more disadvantages associated with them. For instance although round robin scheduling is considered most widely used scheduling algorithm but it has disadvantage of higher context switches if time quantum is too small or it will degenerates to FCFS if time quantum is too large. In case of FCFS turnaround time, waiting time and response time is high. Also in it one process with longest burst time can monopolize CPU. Hence throughput is low. Similarly the real difficulty with the SJF algorithm is, to predict the length of the next CPU request [1]. While it minimizes average wait time, it may penalize processes with high service time requests. That is it leads to starvation of processes. In PS algorithm starvation can happen to the low priority process. The waiting time gradually increases for the equal priority processes.

E. Choice of Scheduling Algorithm

Choice of scheduling algorithms depends on several issues like:

1) Type of the system for which scheduler has to be designed(Multitasking, multiprocessing etc)
2) Whether a process could be preempted or not
3) Metrics which are used to optimize the system performance.

This shows that there is no universal “best” scheduling algorithm. Many operating systems use extended or combinations of the scheduling algorithms stated above.

III. LITERATURE SURVEY

From the above discussion it is clear that CPU scheduling algorithms are generally designed around three important factors-

1) Arrival time of a process in the ready queue

2) Priority of a process in the ready queue

3) CPU burst time of the processes in the ready queue.

In the recent past, a number of CPU scheduling mechanisms have been developed for scheduling allocation of processor.

Depending on these factors the research area is also very wide and could be categorized as follows:-

A. Researches based on dynamic time quantum and nature of processes in the ready queue.

Many researchers had tried to make the static time quantum used in RR scheduling as dynamic. With this technique the real time applications become more responsive as now the time quantum continuously changes with the type of processes in the queue.

Rami J. Matarneh[9] proposed that an optimal time quantum could be calculated by the median of burst times for processes in the ready queue. If this median is less than 25ms, then in such case, the quantum value is modified to 25ms to avoid the overhead of context switch time.

Kiran and etal.[16] proposed a novel approach called Mean-Difference Round Robin Algorithm(MDRR). Their proposed algorithm calculates the mean burst time of all the processes in the ready queue. Then it finds out the difference between a process burst time and calculated mean burst time. This step is repeated for all the processes in the ready queue. Then, it finds out the process having the largest difference value and assigns CPU to it. CPU executes it for one time slice. Once the time slice of the process expires, the next process with the largest difference value is picked up from the ready queue and so on.

Kumkum[19] and etal. suggested another algorithm(ERR) which considers dynamic time quantum in RR scheduling. . In the proposed approach, the time quantum is taken as the ratio of the sum of the burst times to the no. of processes.

Aashna Bisht[22] and etal. had slightly modified the time quantum for the processes that require only a fractional greater time than the allotted time quantum cycle(s) to complete their execution. They again combined dynamicity with RR scheduling.

Lalit Kishore [13] etal. proposed a Median based time quantum based scheduling algorithm. It is a combination of SJF & RR where all the jobs in the queue are first ordered as per their burst time in ascending order and then Round robin is applied for improving the CPU performance.

K.N. Rout [15] and etal. calculated the time quantum using median and highest burst time and then executed the processes as per the new calculated time quantum.

P. Surendra Varma[14] and etal. proposed SRBRR algorithm which assigned the processor to processes with shortest remaining burst in round robin manner using the best possible time quantum. they showed that SRBRR performs better than RR algorithm in terms of reducing the number of context switches, average waiting time and average turnaround time.

Mohammed Abdullah Hassan Al-Hagery[5] proposed a method that takes into account the basics of RR algorithm. The proposed method is called Selective-Round-Robin Quantum of Time (SRRQT). The SRRQ is one of the researches that provide best solution and perfect results to the challenge of RR by using selective time quantum instead of a fixed value.

Sourav Kumar Bhoi[24] and etal. proposed a new effective dynamic RR algorithm SMDRR (Subcontrary Mean Dynamic Round Robin) based on dynamic time quantum where they used the subcontrary mean or harmonic mean to find the time quantum.

B. Researches based on combination of dynamic time quantum with other factors such as priority etc.

Al-Husainy[4] proposed a new algorithm Best Job First(BJF), which mixes the functions of some existing standard algorithms. He combined the very three important scheduling factors of any algorithm viz. priority, arrival time and CPU burst time. In his approach he calculated a new factor (f) for generating a ready queue in that order. His research shows that the process with the highest priority, shortest burst time and the one which is submitted to the system early will be executed first. His results show that BJF is better than FCFS, priority, RR and SJF algorithms. his research has proven a milestone in this area.

Yaashuwanth and R. Ramesh[28] said that a dedicated small processor can be used to reduce the burden of the main processor in assigning for rearranging of processes in the ascending order based on the CPU burst of the process (lower to higher). They proposed an architecture which eliminated the defects of implementing simple round robin architecture in soft real time system by introducing a concept called intelligent time slicing which depends on three aspects they are priority, average CPU burst, and context switch avoidance time. The calculated time slice will be different and independent for each task

Rakesh Mohanty[29] and etal. proposed a new algorithm, known as Priority Based Dynamic Round Robin Algorithm (PBDRR). This approach calculates intelligent time slice for individual processes and changes after every round of execution. The proposed scheduling algorithm is developed by taking dynamic time quantum concept into account. In their algorithm, the shorter processes are given more time quantum so that they can finish their execution earlier. That is their algorithm is the combination of SJF and RR algorithm with dynamic time quantum.

Himanshi Saxena[20] extended the concept of Round Robin algorithm (RR) that included priority of a process. They calculated the factor of precedence 'FP' for each process which determines the order of execution of processes, intelligent service time 'IST' for each process which determines time of execution of process in a single round and execute the processes in RR fashion.

P.Surendra Varma[23] and etal. showed that performance of the RR can be improved by taking mean average of burst times as time quantum. In their research, they proposed a novel scheduling algorithm which uses mean average as a time quantum and also uses the balanced factor (BF) of precedence rather than arrival time to find the order of execution of processes.

H. S. Behera[27] proposed a new variant of Round Robin scheduling algorithm, known as Precedence based Round Robin with Dynamic Time Quantum (PRRDTQ). Their algorithm gives precedence to all processes according to their priority and burst time, then applies the Round Robin algorithm on it. In their proposed approach processes with shorter burst time and higher priority are executed first thus resulting in better turnaround time and better waiting time. Their experimental result showed that the proposed algorithm performs better than algorithm in MRR [28] and PBDRR [29] in terms of CPU performance.

Zena Hussain Khalil[25] and etal**.** have improved RR by proposing a new algorithm which uses the concept of Dynamic quantum time in RR, by taking the Highest Response Ratio Next (HRRN) for each process in each round to select the next process from ready queue In their proposed algorithm, Intelligent Time Slice (ITS) is calculated to give a different time quantum to each process based on the difference between CPU burst time, context switch avoidance time and priority.

Reena Kumari Naik[26] and etal**.** proposed a new hybridized multilevel feedback queue (MLFQ) with intelligent time slice (ITS)**.** The proposed that the processes that are entering into the system are assigned to the first ready queue according to their priority which is decided by using HRRN [25] algorithm and then gradually shifted to the next lower level queues upon expiration of their time slice.

C. Researches based on combination of artificial techniques with the existing scheduling algorithms

Some of the researches even used artificial intelligence in determining dynamic time quantum or priority. A lot of efforts also in this field have been done.

Becchetti[10] and etal. showed that recurrent Neural Network can be used to optimize the number of queues and quantum of each queue of MLFQ scheduler to decrease response time of processes and increase the performance of scheduling.

Deepali Maste[21] proposed a new variant of MLFQ algorithm, in which time slice is assigned to each queue for MLFQ scheduling such that it changes with each round of execution dynamically and neural network is used to adjust this time slice again to optimize turnaround time

In ***Basney***[30] research paper, they carried out a smoothed competitive analysis which is applied to multilevel feedback algorithm.

IV. EFFICIENCY OF EXISTING APPROACHES

From the above discussion it has been clear that an efficient algorithm is one which schedules the processes in a fair manner and also produces results in minimum average waiting time and turnaround time.

It has been discussed in various research papers that time quantum and priority decision is the bottleneck in various CPU scheduling algorithm. What should be the optimal time quantum in round robin algorithm is a difficult question to answer. How a relationship between dynamic priorities in conjunction with other algorithms is another aspect to be considered. The meaning of dynamicity in CPU scheduling algorithms is not an easy task to determine. The questions are numerous but the solution set must satisfy most of the performance criteria and the solution must be universally accepted by most of the operating system.

Few researchers paid attention towards deciding dynamic time quantum, some talk about dynamic priority calculation and some paid attention towards the combination of two. Some says that response time must be the first criteria to be looked upon. Some of the researches even used artificial intelligence in determining dynamic time quantum or priority.

Researches shows that a lot of efforts in this field have been done, and most of the work is done on the basis of calculating dynamic time quantum using measures of central tendency (mean, median). But since these measures suffers from the effect of outliers in the set, so such solutions failed to give optimum result for the set of processes having CPU burst time in which outliers are present. An outlier is generally considered to be a data point that is far outside the norm for a variable. Outliers can have deleterious effects on central tendency measurement. If the outlier is higher, it will lower the mean. If the outlier is lower, it will raise the mean. The mean would get more accurate without the outlier value, which is why they're sometimes removed

Also it is a known fact that the CPU always tries to keep a mix of CPU bound and I/O bound processes [1], so chances of presence of outliers in the list is not an exception. Removing the outlier makes the data more accurate while determining mean or median value for determining time quantum.

So the proposed approach addresses this issue in CPU scheduling algorithm.

V. PROPOSED ALGORITHM (PSSDTQ)

A. Approach

From the above literature review of the CPU scheduling algorithms, it could be easily figure out that there are three major factors attached with each process namely arrival time, burst time and the priority.

So in the proposed approach, a new factor PF (precedence factor) has been suggested to attach with each submitted process. This will combine the effect of all the three basic factors (priority, CPU burst time and arrival time). The equation summarizing this relation is[4]:

*PF = (priority*0.8) + (Burst Time*0.7) + (arrival time * 0.2)* [4] ……………… (1)

Normally the priority factor is the most important factor than the arrival time and burst time in a real time environment. Then CPU burst time can be considered more important factor than the arrival time. Therefore in this equation more weight is given to priority (80%). Also, the rule of thumb is that 80 percent of the CPU bursts should be shorter than the time quantum [1]. Therefore 0.7 weight is assigned to the Burst time, so that at least the target of 70% is reached.

This algorithm uses a combination of two techniques of scheduling namely Priority (PS) & Round Robin (RR).

A precedence factor PF as mentioned in equation (1) is assigned to each process depending on the priority and arrival time of process.

Also it is evident that performance of RR algorithm solely depends upon the size of time quantum. If it is very small, it causes too many context switches. If it is very large, the algorithm degenerates to FCFS. So proposed algorithm solves this problem by taking a dynamic time quantum where the time quantum is repeatedly adjusted according to the remaining burst time of currently running processes. To get the optimal dynamic time quantum, harmonic mean is taken into account. This is done to cancel the effect of the outliers. Equation (2) will explain it:

$$\text{Time Quantum (TQ)} = n/(1/tq_1 + 1/tq_2 + 1/tq_3 + \ldots + 1/tq_n) \quad \text{………………….(2)}$$

where,

n= no. of processes

tq_i = CPU burst time of ith process

B. Algorithm

Since this algorithm is designed around multiple parameters of CPU scheduling algorithm, so it is termed as proportional share scheduler with dynamic time quantum (PSSDTQ)

Input : CPU Burst time, Priority(P_i) and arrival time of each process.

1) *Start*
2) *sort the processes on the basis of burst time and assign number BT to each process such that highest numberis assigned to the shortest job*
3) *Sort the processes on the basis of their arrival time and give number ATT to each process such that the job coming early in the system will get the highest number.*
4) *Calculate Precedence factor (PF) as follows*

$PF_i = 0.8 * P_i + 0.7 * BT_i + 0.2 * ATT_i$

for each process in the queue.

5) *Now processes are ordered such that the process with highest PF will be executed first. If there are any ties in PF values then it will be resolved by considering process with highest priority.*
6) *Once the execution sequence is found, now compute the time quantum using*

$\text{Time Quantum (TQ)} = n/(1/tq_1 + 1/tq_2 + 1/tq_3 + \ldots + 1/tq_n)$

where,

n= no. of processes

tq_i = CPU burst time of ith process

7) *Assign TQ to each process Pi in the queue and allocate CPU in a RR fashion*
8) *Repeat step 6 and 7 until the ready queue becomes empty.*
9) *Stop*

VI. CONCLUSION

It has been found that any kind of simulation for any standard CPU scheduling algorithm has limited accuracy. Therefore it led to a lot of research work in order to design a CPU scheduling algorithm that assure fairness and avoid starvation during allocation of CPU to the processes.

This study shows that

1) Careful attention is required to assure fairness and avoid starvation during allocation of CPU to the processes.
2) CPU must be allocated for that time quantum for which minimum average waiting time and turnaround time is achieved. The best approach for this is to include dynamicity in the CPU scheduling algorithms.
3) This paper proposes a method to improve the RR algorithm and thereby propose a new algorithm which computes the dynamic time quantum on the set of available processes.
4) Also this algorithm gives precedence to those processes which have higher priority and lower burst time and arrival time.

VII. FUTURE SCOPE

The work presented in this paper can be expanded in many directions like:

1) Studying performance in real time applications where tasks have priorities and deadline constraints.
2) Applying scheduling technique on distributed systems.
3) Employing different performance criteria for comparison such as the turnaround time and response time.

REFERENCES

[1] A. Silberschatz, Galvin, and G. Gagne, “Operating System Concepts“, 2009,Wiley Inc.

[2] A. S. Tanenbaum, “Modern Operating Systems” 2009 Prentice Hall(book).

[3] H.S. Behera, Rakesh Mohanty, Jajnaseni Panda, Dipanwita Thakur, Subasini Sahoo “Experimental analysis of a new fair-share scheduling algorithm with waited time slice for real time systems”. Journal of Global Research in Computer Science (ISSN-2229-371X), Volume 2, No. 2, 54-60, February 2011.

[4] Al-Husainy, M.A.F.,”Best-job-first CPU scheduling algorithm” 2007, Inform. Technol. J.,Volume 6: Number 2, Pp: 288-293(pp)

[5] Mohammed Abdullah Hassan Al-Hagery “A selective quantum of time for round robin algorithm to increase cpu utilization”, 2011, International Journal of Computer Information Systems, Vol.3 No.2, 2011

[6] Shlomi Dolev , Avi Mendelson and Igal Shilman,” Semantical cognitive scheduling”, Nov. 2012 Technical Report # 13-02 Research in part by Microsoft and (Fronts).(pp)

[7] Rakesh Kumar Yadav, Anurag Upadhayay , “A fresh loom for Multilevel Feedback Queue Scheduling Algorithm”, July 2012,International Journal of Advances in Engineering Sciences Vol.2, Issue 3.(pp)

[8] Abbas Noon1, Ali Kalakech2, Seifedine Kadry “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average“May 2011, IJCSI International Journal of Computer Science Issues, Vol. 8

[9] Rami J. Matarneh, “Self-Adjustment TQ in RR Algorithm Depending on Burst Time of the Now Running Processes”,2009, American Journal of AS, Vol 6, No. 10.(pp)

[10] Becchetti, L., Leonardi, S. and Marchetti S.A., “Average-Case and Smoothed Competitive Analysis of the Multilevel Feedback Algorithm” 2006, Mathematics of Operation Research Vol. 31, No. pp. 85–108.(pp)

[11] Sanjay Kumar Panda, Enhancing CPU Performance using Sub contrary Mean Dynamic Round Robin (SMDRR) scheduling algorithm, Volume-2, No.12, December 2011.

[12] Sanjay Kumar Panda, An Effective Round Robin Algorithm using Min-Max Dispersion Measure, Volume-2, No.12, December 2011.

[13] Lalit Kishore, Dinesh Goyal, Time Quantum Based Improved Scheduling Algorithm, Volume 3, Issue International Journal of Advanced Research in Computer Science and Software Engineering. ISSN: 2277 128X, 4,April, 2013.

[14] 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, Volume 2, Issue 11, ISSN: 2277 128X , November 2012.

[15] KN Rout, G.Das,B.M. Sahoo,A.K. Agrawalla, ,Improving Average Waiting Time Using Dyanamic Time Quantum,IRAJ,2013.

[16] Kiran, Polinati Vinod Babu, B.B Murali Krishna, Optimizing CPU Scheduling for Real Time Application Using Mean-Difference Round Robin (MDRR) Algorithm,springer,2014

[17] H.S. Behera, Simpi Patel, Bijayalakshmi Panda, A New Dynamic Round Robin and SRTN Algorithm with Variable Original Time Slice and Intelligent Time Slice for Soft Real Time Systems

[18] Supriya Raheja, , Reena Dhadich, Smita Rajpal, An Optimum Time Quantum Using Linguistic Synthesis for Round Robin CPU Scheduling Algorithm, International Journal on Soft Computing (IJSC) Vol.3, No.1, February 2012

[19] Kumkum Pattanayak, Payal Pansari, Subhashree Patra, Subhashree Mohapatra, An Enhanced Round Robin (ERR) Scheduling Algorithm using Dynamic Time Quantum,IJSAA,2013

[20] Himanshi Saxena, Prashant Agarwal, Design and Performance Evaluation of Precedence Scheduling Algorithm with Intelligent Service Time (PSIST), ICCMS, 2012
[21] Deepali Maste, Leena Ragha and Nilesh Marathe, Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling, IJCIT ,2013
[22] Aashna Bisht, Mohd Abdul Ahad, Sielvie Sharma, “Enhanced Round Robin Algorithm For Process Scheduling Using Varying Quantum Precision”, ICRIEST-AICEEMCS, 29th December 2013, Pune India
[23] P.Surendra Varma ,Design and Performance Evaluation of Precedence Scheduling algorithm with Mean Average as Time Quantum (PSMTQ), 2012
[24] Sourav Kumar Bhoi, Sanjaya Kumar Panda and Debashee Tarai,Enhancing CPU Performance Using Subcontrary Mean Dynamic Round Robin (SMDRR) Scheduling Algorithm, JGRCS,2011
[25] Zena Hussain Khalil , Ameer Basim Abdulameer Alaasam , “Priority Based Dynamic Round Robin with Intelligent Time Slice and Highest Response Ratio Next Algorithm for Soft Real Time System”, Global Journal of Advanced Engineering Technologies, Vol2, Issue3-2013
[26] H.S.Behera , Reena Kumari Naik, Suchilagna Parida, A new hybridized multilevel feedback queue scheduling using dynamic time quantum, International Journal of Engineering Research & Technology, Vol.1 - Issue 3 (May- 2012)
[27] H. S. Behera Brajendra Kumar Swain, A New Proposed Precedence based Round Robin with Dynamic Time Quantum (PRRDTQ) Scheduling Algorithm For Soft Real Time Systems, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6,2012
[28] C. Yaashuwanth and R. Ramesh, : A New Scheduling Algorithm for Real Time System, International Journal of Computer and Electrical Engineering (IJCEE), Vol. 2, No. 6,pp 1104-1106, December, 2010.
[29] Rakesh Mohanty, H. S. Behera, Khusbu Patwari, Monisha Dash, M. Lakshmi Prasanna "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.
[30] Basney, Jim and Livny, Miron,“Managing Network Resources in Condor”,2000 9th IEEE Proceedings of the International Symposium on High Performance Distributed Computing, Washington, DC, USA.