ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Analysis of Modified Round Robin CPU Scheduling Algorithm

Mohd Vasim Ahamad
University Women's Polytechnic
Aligarh Muslim University
Aligarh, India

Atiqua Ansari
University Women's Polytechnic
Aligarh Muslim University
Aligarh, India

Uqba Jabeen
University Women's Polytechnic
Aligarh Muslim University
Aligarh, India

Abstract: Operating System is the resource manager of our computer system. And one of most important resource of the computer system is CPU. In multiprogramming operating system, there are several jobs in the main memory. There must be a proper criteria to serve these jobs, this criteria is called CPU scheduling. There are different algorithms to schedule the CPU but each of them have their advantages and disadvantages. In this paper, we have considered mainly Round Robin (RR) Algorithm. This algorithm has the demerit of having high Average Waiting Time (AWT) and high Average Turnaround Time (ATT). The goal is to propose and analyse a CPU Scheduling algorithm in order to reduce the Average Turnaround Time and Average Waiting Time by combining Round Robin Algorithm with Shortest Job first (SJF) Algorithm. To achieve the goal, we have propose two approaches to modify the existing Round Robin Algorithm. In first approach, we have combined the Round Robin Algorithm with Shortest Job First Algorithm and assigned the time quantum as the lowest burst time of first process in sorted (ascending) ready queue. By doing this we are able to reduce Average Turnaround Time but our aim is to reduce the Average Waiting Time also. Therefore, we proposed second approach, in this we have implemented the combination of Round Robin Algorithm with Shortest Job First Algorithm and assign the time quantum with the average of burst time of all the process of the ready queue. By doing this we are able to reduce Average Turnaround Time as well as Average Waiting Time, which was our ultimate goal.

Keywords: Operating Systems, CPU Scheduling, Multiprogramming, Round Robin Scheduling, Shortest Job First Scheduling

I. INTRODUCTION

Operating system is a core part of a computer system which performs variety of tasks to run client application over the system. The CPU scheduling is one of the basic task of an Operating System. All the resources associated with a computer system are needed to schedule before application software can use them. In multiprogramming systems, multiple processes can run simultaneously. Initially, all programs are stored on secondary memory in a job pool. Whenever needed, processes are moved from job pool to main memory. Every process needs CPU and other I/O devices to complete its task. But, process execution and termination is done by CPU only. When CPU is free, it is allocated to a particular process. In the case, multiple processes are competing to get the CPU, scheduling comes into action. Scheduling is a technique to provide a sequence to CPU in which it should execute processes currently in main memory.

Process can be defined as a running program which needs a number of resources such as CPU time, memory, I/O devices, etc. to complete its task [1][4]. In multiprogramming environment, more than one processes are kept in the ready queue, which are waiting to take control over the CPU. As multiple processes are waiting in ready queue to be executed, computer system uses different scheduling algorithms that ensures efficient process execution and reduce waiting time of each process and the most important factor, it also increases the CPU utilization[1][9].

In this paper, we have considered mainly Round Robin (RR) Algorithm. This algorithm has the demerit of having high Average Waiting Time (AWT) and high Average Turnaround Time (ATT). The goal is to propose and analyze a CPU Scheduling algorithm in order to reduce the Average Turnaround Time and Average Waiting Time by combining Round Robin Algorithm with Shortest Job first (SJF) Algorithm.

The remaining sections of this paper is organized as follows: A review of the research work previously carried out in this field is listed in section II. Basic concepts related to multiprogramming and CPU scheduling are presented in section III. In Section IV, we have provided the detailed description of proposed work. Results and discussion are described in section V followed by the conclusion and future work in the last section.

II. LITERATURE REVIEWS

One of the most important components of the computer resource is the CPU. CPU scheduling is the basis of operating systems. Scheduling is the technique used for controlling the order of process which is to be performed by a CPU of a computer. Some of the popular CPU scheduling are First Come First Served (FCFS), Shortest Job First (SJF), Priority Scheduling and Round Robin (RR). Most CPU scheduling algorithms concentrate on maximizing CPU utilization and throughput and minimizing turnaround time, response time, waiting time, and number of context switching for a set of requests.

The authors in [1], designed a new Improved Round Robin (RR) Scheduling Algorithm. This Scheduling algorithms gives better result compare to Round Robin (RR), Improved Round Robin (IRR), Enhanced Round Robin (ERR), Self-Adjustment Round Robin (SARR) [6], FCFS and some other scheduling algorithm in enhancing the CPU performance and its efficiency.

By using Radhe Shyam and Parmod Kumar's proposed algorithm, we are getting better Average Waiting Time, Average Turnaround Time and Context Switch. As authors have taken the ideal cases in calculating the TAT and WT.

The authors in [2], have designed an efficient job scheduling algorithm to minimize processing time of the jobs, which improves the result of non-grouping algorithm. Group based analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a group and how to efficiently find them. Popular notions of grouping include groups with small distances among the group members, dense areas of the data space, intervals or particular statistical distributions. Grouping can therefore be formulated as a multi-objective optimization problem. Grouping gives better results than implementing algorithms separately.

Furthermore, the authors in [3], have got some better performance in terms of processing time than job scheduling on the FIFO algorithm. Also authors have implemented Shortest Job First algorithm in Round Robin algorithm performance analysis. However, allocating large number of jobs to one resource will increase the processing time. Therefore to avoid this situation during job grouping activity, the total number of jobs group should be created such that the processing loads among the selected resources are balanced.

III. Basic Concepts

A. *Operating Systems*

Operating system is a core part of a computer system which performs variety of tasks to run client application over the system. An Operating system is a system software that bridges the gap between user and computer hardware by acting as an intermediary between them. The main purpose of an operating system is to provide an environment in which a user can execute application programs efficiently. An operating system provides different functionalities to serve the request of users such as memory management, file management, device management, CPU Scheduling, process management, protection and security [4].

B. *CPU Scheduling*

The CPU is one of the most important part of computer system and its scheduling is one of the basic task of an Operating System [4][9]. All the resources associated with a computer system are needed to schedule before application software can use them. In multiprogramming systems, multiple processes can run simultaneously. Initially, all programs are stored on secondary memory in a job pool. Whenever needed, processes are moved from job pool to main memory. Every process needs CPU and other I/O devices to complete its task. But, process execution and termination is done by CPU only. When CPU is free, it is allocated to a particular process. In the case, multiple processes are competing to get the CPU, scheduling comes into action. Scheduling is a technique to provide a sequence to CPU in which it should execute processes currently in main memory.

C. *CPU Scheduling Criteria*

CPU scheduling decides which process should take the CPU next. There are a number of scheduling algorithm that can be used as per the requirement. Different scheduling algorithms have different criteria which decide which process from the set is to be selected for execution by CPU. It also differentiates different algorithm from each other [4][7][9]. There are as follows

1) CPU Utilization: It is the time period in which CPU is busy in running different processes. Multiprogramming systems aims to maximize the CPU utilization. That means, try to keep CPU as busy as possible.

2) Throughput: Throughput can be defined as the number of processes completed in a unit of time. Higher is the throughput, more efficient is the computer system.

3) Waiting Time: The set of processes are kept in main memory before allocated to the CPU. As, only one process can take the control of CPU at a time, other processes must wait in the ready queue. Waiting time can be defined as the amount of time spent in the ready queue for the allocation of the CPU.

4) Turnaround Time: After taking the control of CPU, process executes its designated tasks and floats between the CPU and I/O cycle. In between the execution, when the process needs to wait, it is kept into waiting queue. Finally, a process is terminated with CPU cycle. For a process, the amount of time spent from its submission into ready queue to its completion is called as the turn-around time.

D. *CPU Scheduling Algorithms*

In early days of computer evolution, uniprocessor systems were used, which allows to run single process at a time and other processes must wait until CPU is completely freed. After that the CPU can be rescheduled to take up another process. In uniprocessor system, when a running process needs I/O operation to complete, the CPU sits idle and no other process can take the control of the CPU in that time. This reduces the overall CPU utilization. To deal with this demerit, multiprogramming systems are used. The basic goal of multiprogramming system is to maximize the overall CPU utilization by running some processes at all times. In order to achieve this, a process is executed until it must wait for I/O operation and CPU is freed.

In multiprogramming systems, system tries to use this time productively by allocating the CPU to another process which are kept in main memory and waiting for CPU. This CPU allocation pattern continues. CPU Scheduling is a basic functionality of an operating system. There are many CPU scheduling algorithms exist namely First Come First Serve (FCFS), Shortest Job First (SJF), Shortest Remaining Job First (SRJF), Priority Scheduling, Round Robin (RR) and Multilevel Feedback Queue Scheduling [1][4][9]. We discussed only Shortest Job First and Round Robin CPU scheduling algorithm, which are the foundation of this research.

1) Shortest Job First CPU Scheduling: There may be a set of processes waiting in the ready queue for CPU allocation. In Shortest Job First (SJF) algorithm, the process which has the shortest CPU burst will be allocated to the CPU. In the case, if two or more processes are having same

CPU burst time, to break the tie, system uses First Come First Serve scheduling. It maximizes task throughput, by running tasks which take less time to complete first, we can complete more tasks in a given amount of time. Therefore, the Shortest Job First scheduling algorithm decreases the waiting time of the processes[1][2][8].

2) Round Robin CPU Scheduling: The Round Robin scheduling algorithm is similar to First Come First Serve scheduling algorithm, but, to provide time sharing capability to the system, preemption is added. It associates a small amount of time to FCFS ordered processes. This time unit is called as time quantum. A time quantum generally ranges from 10 to 100 milliseconds. To implement the RR scheduling, the ready queue is treated as a circular queue. The CPU scheduler takes the first process from the ready queue and allocates the CPU to it for one time quantum. When time quantum expires, next process in the queue is taken from the ready queue and allocated to CPU. If CPU burst time of process is remaining after time quantum expiry, it is added to the tail of the queue. This procedure continues until ready queue is empty [1][5][8].

IV. PROPOSED WORK

A. Problem Statement

Generally, the Round Robin Scheduling algorithm has the demerit of having high Average Waiting Time (AWT) and high Average Turnaround Time (ATT). To maximize the CPU utilization, minimize the average waiting time & average turn-around time, we proposed and analyzed a modified Round Robin scheduling algorithm with dynamic time quantum by combining Round Robin Algorithm with Shortest Job first (SJF) Algorithm. To achieve the goal, we have propose two approaches to modify the existing Round Robin Algorithm.

B. Method 1

In our first approach, we used the minimum CPU burst time, calculated using SJF scheduling as the time quantum. Using minimum CPU burst time as time quantum results reduction in average turnaround time. Basic steps of approach 1 are as follows:

1. Enter n processes in the ready queue Q with their burst times Bi where 1≤ i ≤ n
2. Arrange the burst time of all processes in ascending order
3. Repeat 4, 5 and 6 while Ready Queue becomes empty.
4. Pick the first ascending order sorted process from the ready queue and allocate the CPU to it for a time interval of up to one time quantum.
5. If the remaining CPU burst time of the currently running process is less than one time quantum then allocate CPU again to currently process for remaining CPU burst time. After completion of execution, remove it from ready queue and go to step 3.
6. Remove the currently running process from the ready queue and put it at the tail of the Ready Queue.

C. Method 2

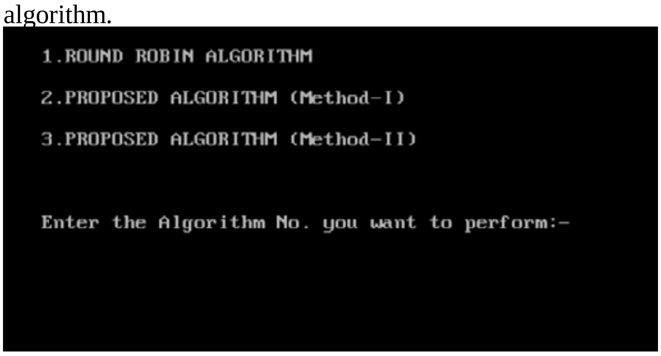
In approach 1, we were able to reduce only the Average Turnaround Time (ATT).Therefore, in our proposed approach 2, we have reduced the Average Turnaround Time (ATT) as well as Average Waiting Time (AWT) by combining Round Robin (RR) algorithm with Shortest Job First (SJF) algorithm but this time we have assigned the time quantum with the burst time of average of all the processes in the ready queue. Basic steps of approach 2 are as follows:

1. Enter n processes in the ready queue Q with their burst times Bi where 1≤ i ≤ n
2. Calculate the average burst time of the processes in ready queue.
3. Repeat 4, 5 and 6 while Ready Queue becomes empty.
4. Take the average of burst time of all the processes in the waiting queue and allocate the CPU to it for a time interval of up to one time quantum.
5. If the remaining CPU burst time of the currently running process is less than one time quantum then allocate CPU again to currently process for remaining CPU burst time. After completion of execution, remove it from ready queue and go to step 3.
6. Remove the currently running process from the ready queue and put it at the tail of the Ready Queue.

V. RESULTS AND DISCUSSION

A. Practical Setup

We have used Turbo C++ Compiler for writing the codes of C language of all algorithm which we have implemented, that is, Round Robin (RR) Algorithm, Proposed Algorithm (Method-I) and Proposed Algorithm (Method-II). After execution of implemented algorithm, here is the snapshot of steps of Result Analysis with respective Gantt chart of all algorithm.



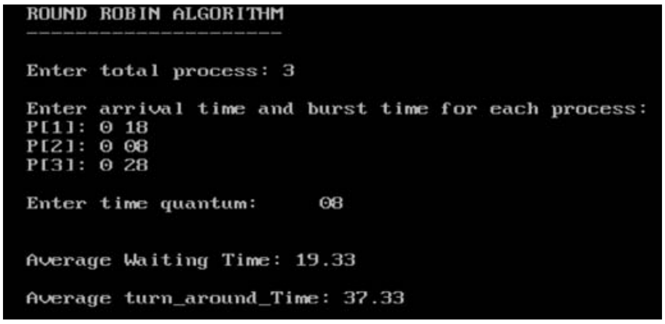
```
1.ROUND ROBIN ALGORITHM

2.PROPOSED ALGORITHM (Method-I)

3.PROPOSED ALGORITHM (Method-II)


Enter the Algorithm No. you want to perform:-
```

Fig. 1 Home Screen



```
ROUND ROBIN ALGORITHM
---------------------

Enter total process: 3

Enter arrival time and burst time for each process:
P[1]: 0 18
P[2]: 0 08
P[3]: 0 28

Enter time quantum:      08


Average Waiting Time: 19.33

Average turn_around_Time: 37.33
```

Fig.2 Result Analysis of Round Robin Algorithm

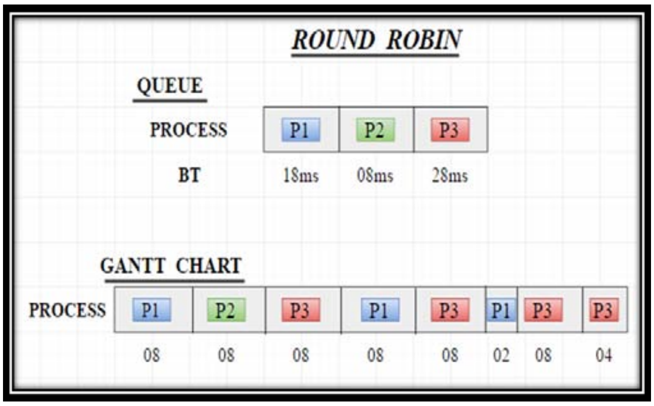
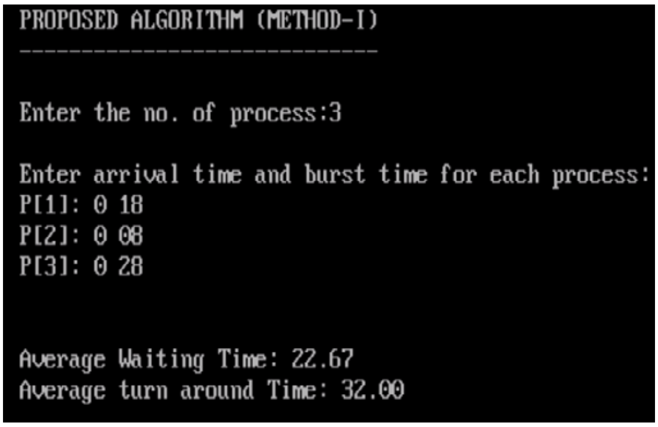


Fig.3 Gantt chart of Round Robin Algorithm



```
PROPOSED ALGORITHM (METHOD-I)
-----------------------------

Enter the no. of process:3

Enter arrival time and burst time for each process:
P[1]: 0 18
P[2]: 0 08
P[3]: 0 28


Average Waiting Time: 22.67
Average turn around Time: 32.00
```

Fig.4 Result Analysis of Proposed Algorithm (Method-I)

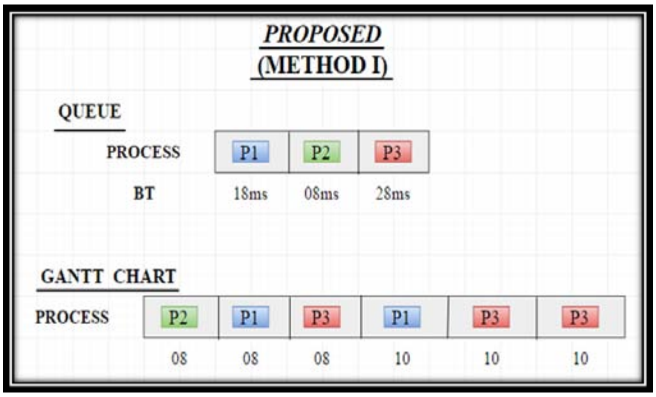
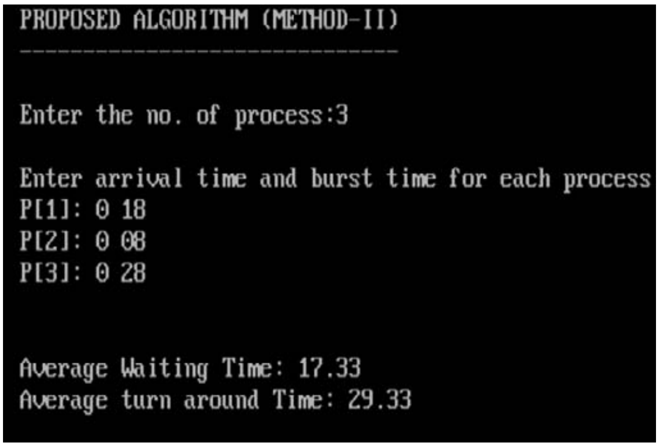


Fig.5 Gantt chart of Proposed Algorithm (Method-I)



```
PROPOSED ALGORITHM (METHOD-II)
------------------------------

Enter the no. of process:3

Enter arrival time and burst time for each process
P[1]: 0 18
P[2]: 0 08
P[3]: 0 28


Average Waiting Time: 17.33
Average turn around Time: 29.33
```

Fig.6 Result Analysis of Proposed Algorithm (Method-II)

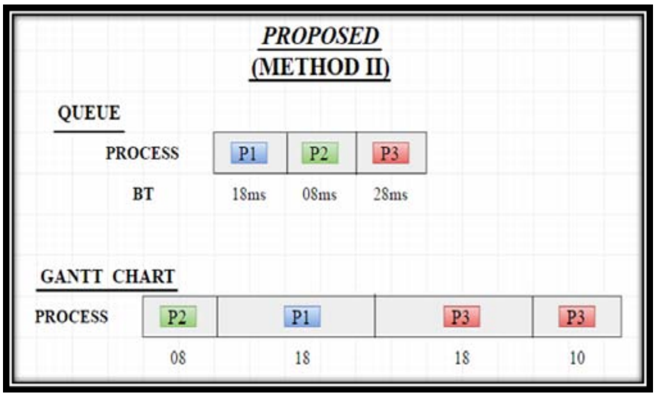


Fig.7 Gantt chart of Proposed Algorithm (Method-II)

B. *Performance Evaluations*

To measure the performance of all algorithm which we had implemented in our project, that is, Round Robin (RR) Algorithm, Proposed Algorithm (Method-I) and Proposed Algorithm (Method-II), we have calculated and compared the following performance measures:

- Average Waiting Time (AWT)
- Average Turnaround Time (ATAT)

1) Avearge Waiting Time: To The CPU Scheduling algorithm does not affect the amount of time during which a process execute or does I/O, it affects only the amount of time that a process spends in the ready queue .Waiting Time (WT) is the sum of the periods spent waiting in the ready queue.

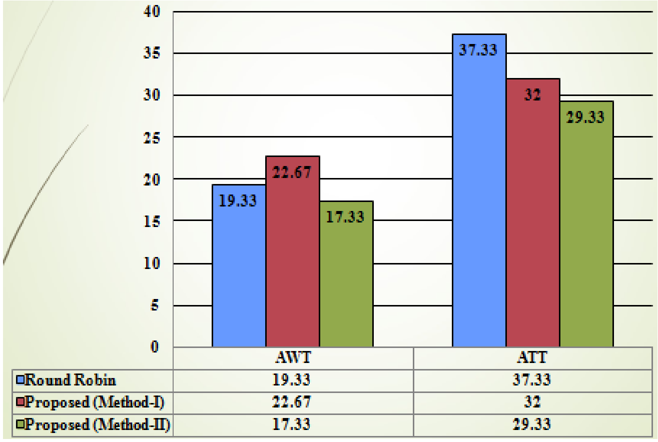
WT=TAT-BT

AWT=Sum of WT of all processes/No. of processes

2) Average Turnaround Time: The interval from the time of submission of a process to the time of completion is the Turnaround Time (TT). It is the sum of the periods sent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

TAT=CT-AT=BT+WT

ATAT=Sum of TT of all processes/No. of processes



	AWT	ATT
■Round Robin	19.33	37.33
■Proposed (Method-I)	22.67	32
■Proposed (Method-II)	17.33	29.33

Fig.8 Comparison of evaluation of all algorithms

VI. CONCLUSION

To achieve, what we have promised in our problem statement, we have implemented Round Robin (RR) Algorithm and our Proposed Algorithm (Method-I) and Proposed Algorithm (Method-II). Left-hand-side of above graph is Average Waiting Time (AWT) and right-hand-side of above graph is Average Turnaround Time (ATT).

After implementation of above mentioned algorithms, as in fig. 8, we found much improvement in Average Waiting Time (AWT) via Proposed Algorithm (Method-II) and Average Turnaround Time (ATT) via Proposed Algorithm (Method-I).

In this paper, we have modified the Round Robin scheduling Algorithm by using the dynamic time quantum. The contributions made in this paper are as follows:

- We analyzed the performances of Round Robin scheduling algorithm and both of the proposed methods.
- In approach 1, we have taken the lowest CPU burst time of the processes in the ready queue as the time quantum. This approach results reduction in average turnaround time only.
- To deal with this, we proposed approach 2, in which we used the average of CPU burst times of all processes in the ready queue.
- We also compared the average waiting time and average turnaround time of all the three algorithms.

Thus, by analyzing the results, we conclude that the overall performance of approach 2 is far better than the other two algorithms.

VII. REFERENCES

[1] Radhe Shyam, Sunil Kumar Nandal, “Improved Mean Round Robin with Shortest Job First Scheduling” in International Journal of Advanced Research in Computer Science and Software Engineering.

[2] Qudsia Mateen, Ujala Niazi, Marwah, Sobia Khalid, “Grouping Based Job Scheduling Algorithm Using Priority Queue, Shortest Job First, Round Robin and First Come First” in International Journal of Computer and Communication System Engineering (IJCCSE), Vol. 2 (1), 2015, 139-142.

[3] Radhe Shyam, Parmod Kumar, “Improved Round Robin with Shortest Job First Scheduling” Extension Lecturer (Computer Science) Govt. College for Women, Hissar, Haryana, India.

[4] Abraham Silberschatz, Peter Baer Galvin,Greg Gagne, Operating System Concepts, sixth edition.

[5] 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.

[6] 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.

[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] Maj. Umar Saleem Butt and Dr. Muhammad Younus Javed, “Simulation of CPU Scheduling Algorithms”,0-7803-6355-8/00/$10.00@2000 IEEE.

[9] Andrew S. Tanenbaum, Albert S. Woodhull, “Operating Systems Design and Implementation”, Second Edition.