Achievers J. Sci. Research
ISSN Online: 2504-0141
volume 2, Issue 2, December 2019 - p. 31 - 38
ACHIEVERS JOURNAL OF SCIENTIFIC RESEARCH
Open Access Publications of Achievers University, Owo.
Available online at www.achieversjournal.org

An Improved Time Varying Quantum Round Robin CPU Scheduling Algorithm

*S. S. Olofintuyi[1], T. O. Omotehinwa[1], E. O. Oyekanmi[1] and E. A. Olajubu[2]
[1]Department of Mathematical Sciences, Achievers University Owo, Nigeria
[2]Department of Computer Science, Obafemi Awolowo University, Ile-Ife, Nigeria

*Corresponding author: olofintuyi.sundaysamuel@gmail.com

Abstract

Round Robin (RR) algorithm has been regarded as impartial algorithm this is because it uses the same quantum time for all processes on the queue irrespective of their burst time. Questions on optimal time quantum to be used by RR and Shortest Job First (SJF) has also been on the mind of researchers. This is because time quantum determines the performance of the algorithm. If the time quantum assigned is relatively high, it may lead to First Come First Serve while high context switch is obtained with low time quantum. This study proposes An Improved Time Varying Round Robin Algorithm (ITVRR).The proposed algorithm was benchmarked with the following algorithms using Gantt chart: Revamped Mean Round Robin (RMRR), Round Robin, First Come First Serve (FCFS) and SJF. The metrics used to benchmark the algorithms are Average Waiting Time (AWT), Average Turnaround Time (AVT) and Context Switch (CS). The result obtained after the experiment suggests that the proposed ITVRR tends to improve context switch and turnaround time compared to RR and RMRR.

Keywords: Time Varying Quantum, CPU scheduling, Round Robin, Gantt chart

INTRODUCTION

Multiprogramming involves running two or more processes simultaneously. It is one of the key areas of operating system. Central Processing Unit (CPU) scheduling comes in place when there are various processes in the memory to be executed, out of these processes; the operating system decides which one to run first. If the CPU is not well managed, it can result to inefficiency of the system. Utilization of the CPU can be maximized if the processes are allowed to be attended to by switching the CPU among the processes (Amar, Sandipta and Sanjay, 2015). The main goal of switching CPU among processes is to reduce the context switch, waiting time, turnaround time and finally increase the utilization of the CPU (Silberschatz, 2005). One of the onuses of Operating System is to manage different processes in memory such that the CPU utilization is optimized. In order to maximize CPU utilization, processes are required to be scheduled in an efficient manner so that maximum numbers of processes are serviced by the CPU. Also, for maximum usability, CPU scheduling has been adopted by many researchers. This work aim at improving the efficiency of CPU by proposing varying time quantum and then benchmarked with RR, RMRR, FCFS and SJF. Section II discusses the standard algorithms used in CPU scheduling. Section III discusses the criteria used to evaluate the performance of a CPU. Section IV discusses the review of literature while section V discusses the proposed algorithm. Experimental results and discussion were done in section VI and section VII respectively while section VIII covers the conclusion.

CPU SCHEDULING ALGORITHM

The conventional CPU scheduling algorithms are FCFS, SJF, RR and priority algorithm. FCFS algorithm operates by attending to the first process on the queue. That is, processes are been attended to base on their arrival. SJF operates by attending to the process that has the shortest burst time. Burst time is the time required for a process to finish executing it task. RR operates by using a quantum time. Quantum time defines the time allotted to CPU to attend to processes on the queue. If in any scenario, the burst time of a process is more than the quantum time, after the CPU have attended to the process, the process is sent back on the waiting queue. It means, the CPU only share it allotted quantum time with all the processes equally. In priority scheduling algorithm, the efficiency of the system is not considered, the CPU only attends to processes that has the highest priority. Processes would have been prioritized before they arrived the waiting queue.

CPU SCHEDULING CRITERIA

For the aforementioned algorithms, various metrics have been used to measure their performance. Such metrics include context switch, average turnaround time, average waiting time, throughput, CPU utilization and response time.

1. Context switch: This has to do with storing and restoring context preempted process in order for the execution to start from the same position.
2. Throughput: This defines the number of processes the CPU complete per unit time.
3. Turnaround time: This defines the time taken by the CPU to execute a process
4. Waiting time: It defines the total number of time that the process waits on the queue
5. Response time: it discusses the very first time the process accesses the CPU
6. CPU

Utilization: utilization of the CPU discusses the usage of the CPU.

An efficient scheduling algorithm should have the following:

1. Minimum context switch
2. Minimum turnaround time
3. Minimum waiting time
4. Minimum response time
5. Maximum CPU utilization
6. Maximum throughput.

REVIEW OF LITERATURE

Sarvesh et al. 2018) presented an algorithm that uses round robin and shortest job first in order to reduce the turnaround time and waiting time. In their work, a fixed quantum time was used for all the processes. Whenever a process is unable to finish it task before the quantum time lapse, the process is placed at the end of the ready queue and the next process with shortest burst time takes up the

CPU but if the process finishes it task before the quantum time lapse, the CPU can then take up another process with the shortest burst time for execution. At the end of their experiment, the work was benchmarked with round robin, short job first using turnaround time, waiting time and context switches. The proposed algorithm performed better than round robin, shortest job first using the metrics aforementioned.

Bhavin and Manoj (2018) presented another approach for scheduling algorithm of round robin using dynamic quantum time in cloud environment. In their methodology, the quantum time was obtained by taking the mean and median of all the burst time of the processes. The mean is considered by taking the average of all the burst time while the median was considered if the result was odd, them the middle number will be the median otherwise, if the result is even, then the median will be the mean of two central numbers. The round robin algorithm used was derived from MRRA algorithm and SRBRR algorithm. After the experiment, the result derived was compared to RR and MRRA in terms of AWT, AVT and CS.

Ahmed, et al. (2018) presented a novel method based on priority for enhancement round-robin scheduling algorithm. The authors proposed an algorithm to optimize the performance of CPU scheduling. They suggest in their work, that all new processes into the queue that has the lowest burst time be prioritized over other processes on the queue. Immediately the burst time of the process is equal to the quantum time, rescheduling should be done to give a new priority to a new task. The proposed algorithm was compared to the traditional round robin with respect to average waiting time and average turnaround time.

LaxmiJeevani, Madhuri and Sarada (2018) proposed an improvised round robin scheduling algorithm and comparison with existing round robin CPU scheduling algorithm. The objective of the proposed algorithm is to reduce the average waiting time and also reduce the average turnaround time so that they can improve on the existing round robin algorithm.

Sushruta, et al, (2017) presented an optimized round robin scheduling algorithm. In their work, the author(s) adopted the normal approaches of round robin in the first cycle after which the CPU is allocated to the smallest burst time of all the processes. Two different scenarios were considered which includes processes with arrival time and processes with zero arrival time. At the end of their research work, the proposed algorithm performed better than the traditional round robin in terms of waiting term and turnaround time.

Sachin, et al. (2016) proposed a revamped algorithm where a round robin CPU scheduling algorithm was used. In their work, seven processes were used with arrival time and burst time respectively. The proposed algorithm produces minimal context switch, average waiting time and average turnaround time compared to RR algorithm. Also, in their work, two queue(s) were introduced in which one stands for ready queue and the other pre-ready queue. The authors combined both RR and FCFS algorithm in their proposed model; the average burst time for all the processes was used as the quantum time. After the experiment, the result was compared to (RR), FCFS, and Shortest Job First (SJF). The proposed model performed better in Response time (RT), than FCFS, SJF and RR.

Pandaba and Prafulla (2016) presented another algorithm for resources allocation in cloud computing. The methodology uses two registers where one was used to store the remaining burst time of the processes and the other register was used to store the average burst time. Round robin algorithm was adopted in their work. The quantum time changes in which it changes to the burst time of the processes. The burst time of the first process in the queue is always the quantum time for the first process while the average of the burst time of all other processes are taken to determine the quantum time. The work was implemented in MATLAB and was bench-marked against the traditional RR algorithm. After the experiment, their proposed algorithm performed better in turnaround time and waiting time.

Amar, Sandipta and Sanjay (2015) proposed an optimized round CPU scheduling algorithm with dynamic time quantum. In their work, the burst time was arranged in ascending order and 25 were taken as the static quantum time for round robin. The problems used were subdivided into two group based on their arrival time of processes with zero arrival time and processes without zero arrival time. The authors also used double time quantum immediately after the first cycle. The proposed model Dynamic Average Burst Round Robin (DABRR) was compared with the following algorithm: RR, IRRVR, SARR, RP-5, MRR, and DQRRR in both scenario of zero arrival time and non-zero arrival time. The proposed model performed better than the previous algorithm in term of context switch, waiting time and turnaround time.

Manish and Faizur (2014) proposed an Improved Round Robin Scheduling Algorithm with Varying Time Quantum (IRRVQ). The authors combine two out of the four algorithms in their proposed model. The two- algorithm proposed are round robin and shortest job first. Only one queue was used in their work, also the processes were arranged in the ascending order of their burst time respectively. The quantum time changes with respect to the burst time of each process on the queue. The work also considers two scenarios with zero and non-zero arrival time. The experimental result shows that IRRVQ perform better than the traditional RR in term of average waiting time and average turnaround time.

Abdulrazaq, Saleh and Junaidu (2014) also presented a New Improved Round Robin CPU Scheduling algorithm. Two queues were used which are ready queue and arrival queue in the round robin algorithm. The quantum time was obtained by taking the average of the process burst time. During execution of the process, if the quantum time finishes before the execution of the process, the CPU checks whether the burst time is less than or equal to the quantum time. If it does, then the CPU is reallocated to the same process but if it does not, the process is sent back to the arrival queue while the CPU attends to another process in the ready queue. The experimental result shows that SJF has the minimal waiting time and turnaround time while RR and Longest Job First with Combination Burst Time (LJF+CBT) produced minimal context switching and average response time. The proposed model only out-performed RR and IRR in term of AWT, AVT and CS.

PROPOSED ALGORITHM (ITVRR)

The proposed algorithm for the implementation of ITVRR is listed below. Figure 1 shows the flowchart for the proposed algorithm

Step 1: Start

Step 2: Create pre-queue

Step 3: Create ready queue

Step 4: Create post queue
Step 5: Arrival of processes to pre-queue
Step 6: Load all processes to ready queue
Step 7: In the ready queue, apply the following
Step 8: Time quantum = 50 percentile of burst time of processes in the ready queue
Step 9: CPU attends to the first process in the ready queue
Step 10: If burst time of process equals to quantum time, then calculate AWT, AVT and CS
Step 11: If burst time not equal to quantum time, then load process to post queue
Step 12: Use dynamic quantum time with respect to burst time of processes in the post queue.
Step 13: If burst time of processes in the post queue is equal to quantum time, then calculate AWT, AVT and CS
Step 14: If ready queue is not empty GOTO Step 6
Step 15: END

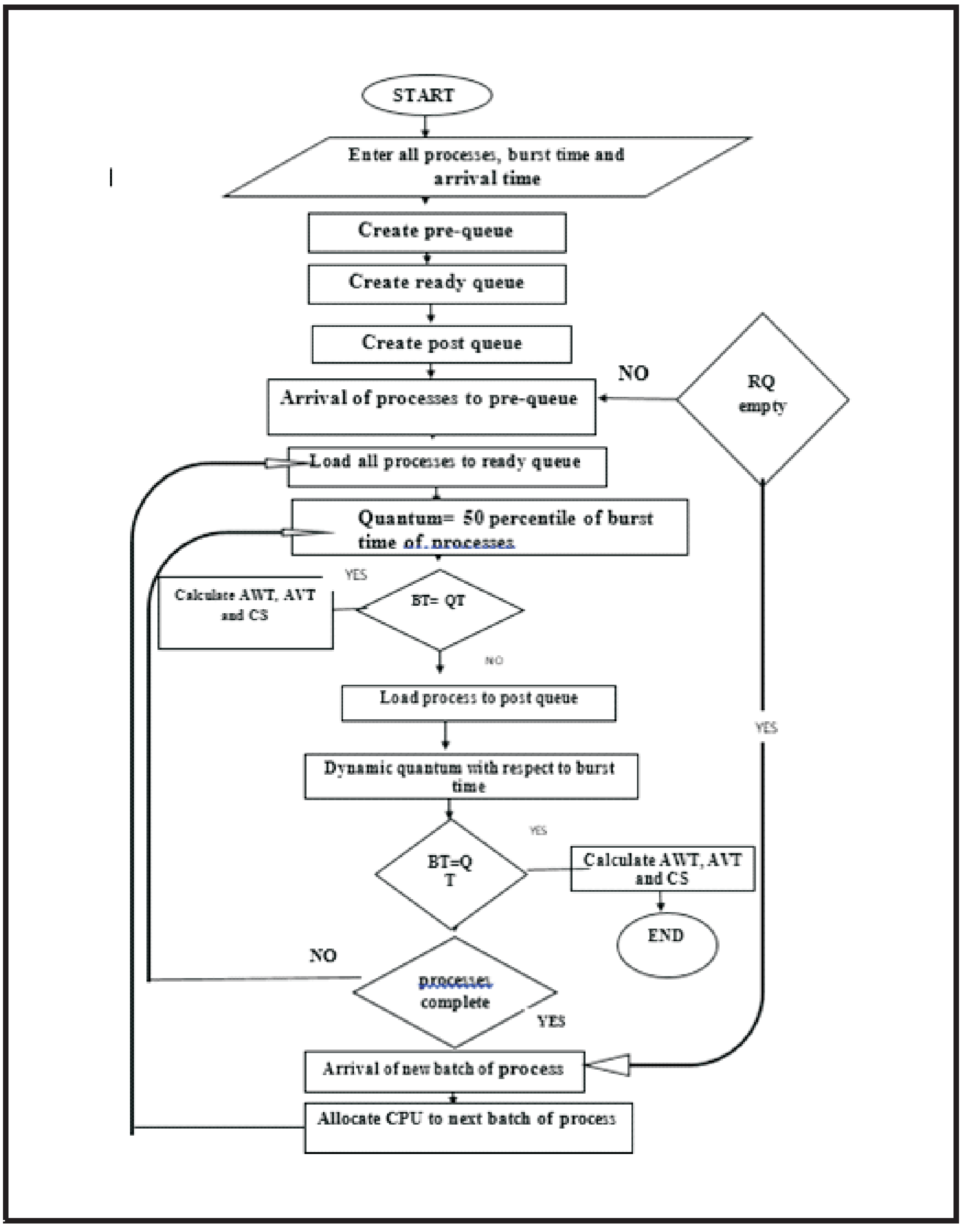
EXPERIMENTAL RESULTS

Table 1 shows seven different processes with their arrival time and burst time.

Table 1: Processes with arrival time and burst time

Processes	Arrival time	Burst time
P0	0	14
P1	2	58
P2	4	18
P3	5	30
P4	1	28
P5	6	46
P6	3	7

Figure 1: Flowchart of the proposed algorithm



P0	P4	P1	P6	P2	P3	P5

0 14 42 100 107 125 155 201

Figure 2: Depicts FCFS Gantt chart

P0	P6	P2	P4	P3	P3	P1

0 14 21 39 67 97 143 201

P0	P4	P1	P6	P2	P3	P0	P5	P4	P1	P6	P2	P3	P0

0 5 10 15 20 25 30 35 40 45 50 52 57 62 66

P5	P4	P1	P2	P3	P5	P4	P1	P2	P3	P5	P4	P1	P3

66 71 76 81 86 91 96 101 106 109 114 119 122 127 132

P5	P4	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P5	P1

132 137 142 147 152 157 162 167 172 177 182 187 188 193 201

Figure 4: Depicts RR Gantt chart

P0	P4	P1	P6	P2	P3	P5	P6	P4	P1	P6	P2	P3	P3	P5	P5	P1

0 2 4 6 8 10 12 14 26 52 79 84 100 127 128 155 172 201

P0	P4	P1	P2	P3	P5	P6	P1	P5

0 14 42 72 90 120 150 57 185 201

Table 2: Depicts performance of some existing and proposed algorithms

Algorithm	Avg. waiting time	Avg.Turnaround time	Number of switch
FCFS	74.57	103.28	6
SJF	51.42	80.14	6
RR	99	127.7	42
RMRR	77.28	106	14
ITVRR	86.8	93	8

The following formula were used to calculate the performance metric

For Waiting time = Turnaround Time - Burst Time

For turnaround time = Burst time + Waiting time Or Completion time – Arrival Time

DISCUSSION

From the experiment conducted using Gantt chart, five different algorithms were used and the following metrices were used to evaluate them. AWT, AVT and number of context switches. The AWT, AVT and context switch for FCFS were 74.57, 103.28 and 6 respectively. For SJF, the AWT was 51.42, AVT was 80.14 and number of context switch was 6. Also, for RR, 99, 127.7, and 42 were the AWT, AVT and number of context switch. Futhermore, the AWT, AVT and context switch for RMRR were 77.28, 106, and 14 respectively. For the proposed ITVRR algorithm, AWT, AVT and number of context switch were 86.8, 93 and 8 respectively. Figure 7, 8 and 9 depicts the chart for all the algorithms using AVT, AWT and number of context switch as an evaluation metric.

CONCLUSION

In this study, a new approach was introduced for CPU scheduling called An Improved Time Varying Round Robin Algorithm (ITVRR). The approach was an improvement on the existing RR algorithm. ITVRR algorithm was compared with other algorithms such as FCFS, SJF, RR and RMRR for CPU scheduling. From the results obtained, it can be observed that ITVRR outperformed RR and RMRR algorithms with respect to AVT and CS but RMRR algorithms only outperformed our proposed model only in AWT. Also, ITVRR outperformed FCFS with respect to AVT.

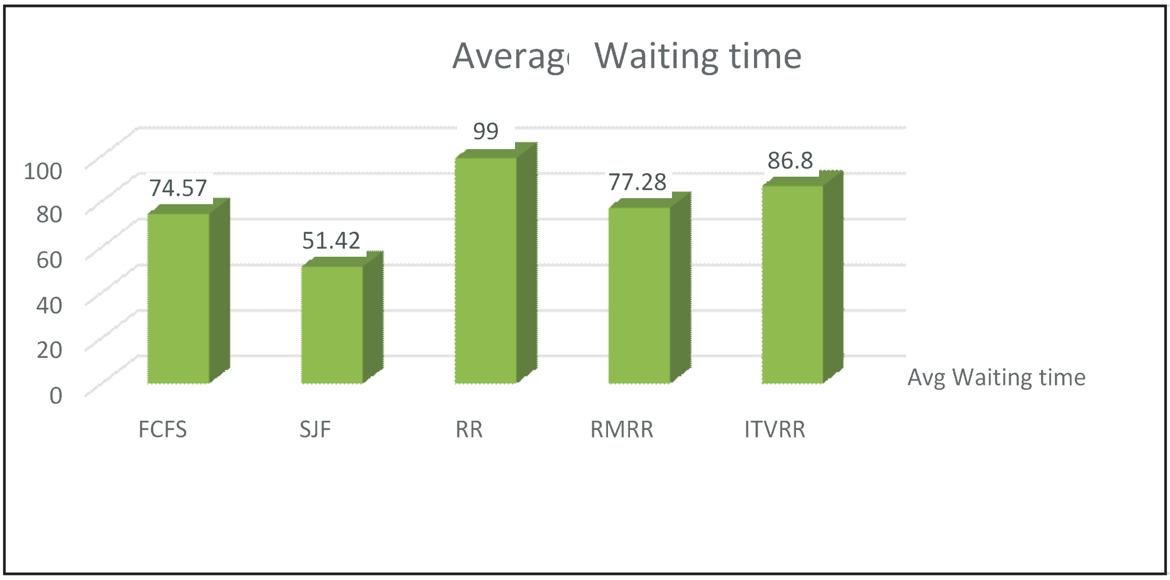


Figure 2: Comparison of FCSF, SJF, RR, RMRR and ITVRR using Avg. waiting time

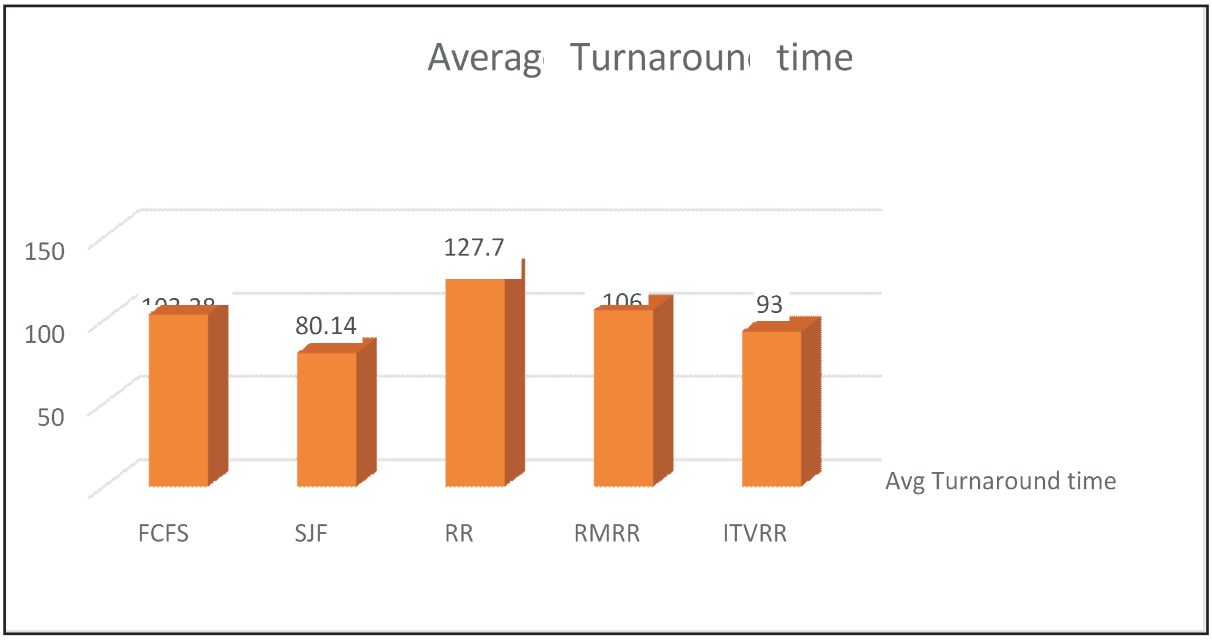


Figure 3: Comparison of FCSF, SJF, RR, RMRR and ITVRR using Avg. turnaround time

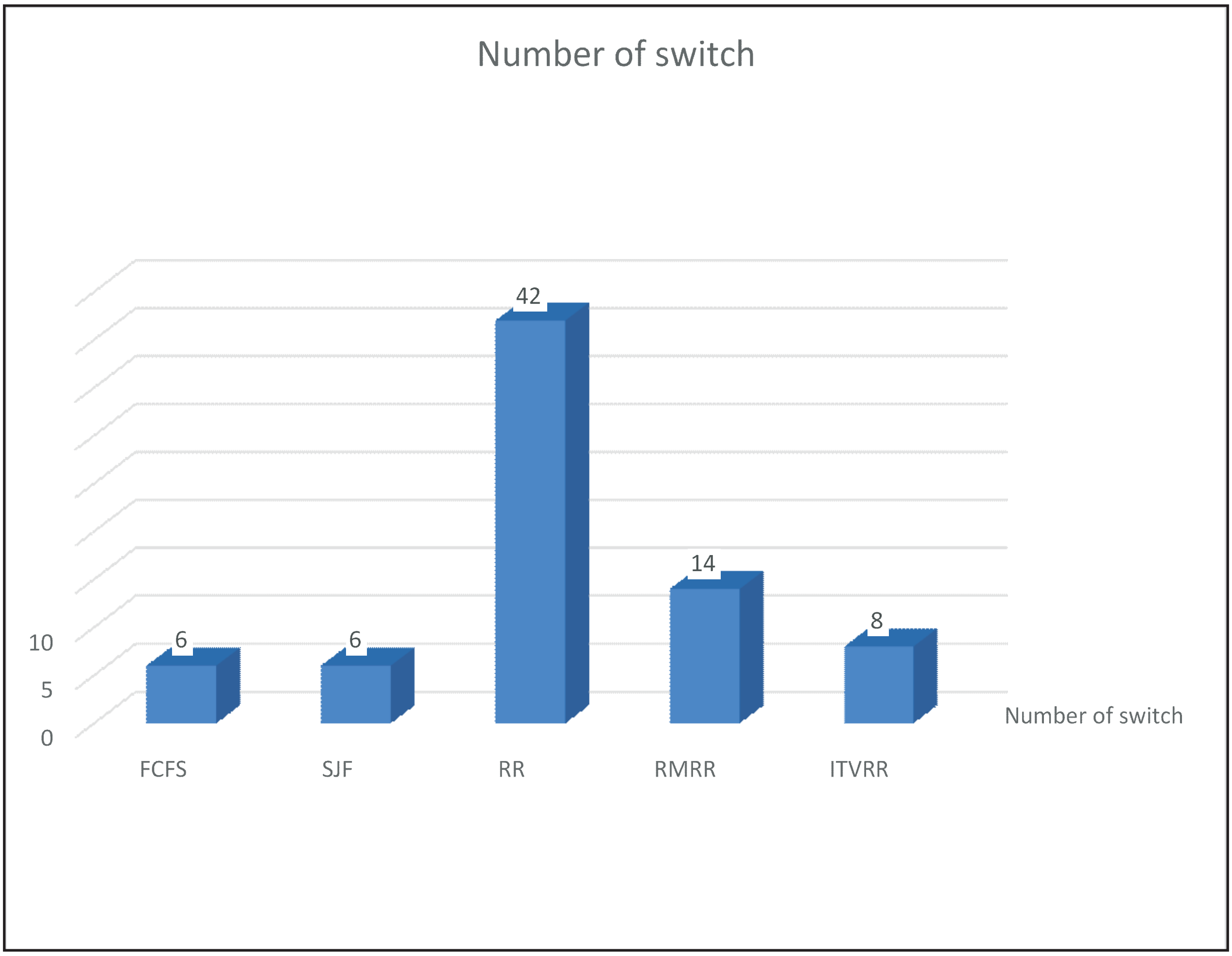


Figure 4: Comparison of FCSF, SJF, RR, RMRR and ITVRR using context switch

REFRENCES

Ahmed, S., A, Hadeel, M., T, and Khalid, W., A. (2018). Novel Method Based on Priority for Enhancement Round-Robin Scheduling Algorithm. Journal of Theoretical and Applied Information Technology. Vol.96, Issue 13.

Bhavin, F., and Manoj, P. (2018). Dynamic Time Quantum Approach to Improve Round Robin Scheduling Algorithm in Cloud Environment. International Journal of Scientific Research in Science, Engineering and Technology. Vol.4, Issue 4

Sarvesh, K., Gaurav, K., Komal, J., and Aditi, J. (2018). An approach to reduce turnaround time and waiting time by the selection of round robin and shortest job first algorithm. International Journal of Engineering & Technology, Vol.7, Issue 2, pp. 667-672

LaxmiJeevani, M., Madhuri, T., and Sarada, D. (2018). Improvised Round Robin Scheduling Algorithm and Comparison with Existing Round Robin CPU Scheduling Algorithm. IOSR Journal of Computer Engineering. Journal of Computer Engineering. Vol. 20, Issue 3.

Sushruta, M., Soumya, S., Sunil, M., and Brojo, K. M. (2017). CPU Scheduling using an Optimized Round- Robin Scheduling Technique. International Journal of Recent Research Aspects, Vol. 4, Issue 3, pp. 221- 225

Sachin, K., Piyush, P., Singh, P. T., and Prashant, M. (2016) A Revamped Mean Round Robin (RMRR) CPU Scheduling Algorithm. International Journal of Innovative Research in Computer and Communication Engineering. Vol. 4, Issue 4.

Pandaba, P., and Prafulla, B. (2016). Modified Round Robin Algorithm for Resource Allocation in Cloud Computing. International Conference on Computational Modeling and Security. *Procedia Computer Science* 85 (2016) 878 – 890

Amar, R., Sandipta, K. S., and Sanjay, K. S. (2015) an Optimized Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum. International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 5, No.1.

Abdulrazaq, A., Saled, E, A., and Junaidu, B.S. (2014). A New Improved Round Robin (NIRR)

CPU Scheduling Algorithm. International Journal of Computer Applications. Vol. 90 No 4, pp. 24 - 35

Manish, K. M., and Faizur, R. (2014). An Improved Round Robin CPU Scheduling Algorithm With Varying Time Quantum. International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.4, No.4

Silberschatz, P. B., Galvin, E., and Gagne, G. (2005) Operating System Concepts, John Wiley and Sons Inc, pp 15.1-167