A Review on the CPU Scheduling Algorithms: Comparative Study

Shahad M. Ali, Razan F. Alshahrani, Amjad H. Hadadi, Tahany A. Alghamdi, Fatimah H. Almuhsin, Enas E. El-Sharawy
Computer Science Department, College of Science and Humanities, Imam Abdulrahman Bin Faisal University, P.O.Box 31961, Jubail, Saudi Arabia

Abstract
CPU is considered the main and most important resource in the computer system. The CPU scheduling is defined as a procedure that determines which process will enter the CPU to be executed, and another process will be waiting for its turn to be performed. CPU management scheduling algorithms are the major service in the operating systems that fulfill the maximum utilization of the CPU. This article aims to review the studies on the CPU scheduling algorithms towards comparing which is the best algorithm. After we conducted a review of the Round Robin, Shortest Job First, First Come First Served, and Priority algorithms, we found that several researchers have suggested various ways to improve CPU optimization criteria through different algorithms to improve the waiting time, response time, and turnaround time but there is no algorithm is better in all criteria.
Key words:
CPU Scheduling Algorithms; RR; FCFS; SJF; Priority.

1. Introduction

Technological development covered all aspects of human life, such as mobile phones and computers, people can shop, study, book appointments, games, and more. Humans will not be able to use these devices and manage them easily due to the hardware's complexity, so the operating systems appeared to solve these problems. It can be defined as an intermediary between the user and the computer hardware to facilitate the computer system's management and control. Operating systems provide many services to the user and the system on the user side it provides user interfaces and helps to implement programs, manage files, and exchange information with other computers, and on the system side, it provides the ability to allocate resources to multiple users and protect system resources [1].

In multi-processing systems, there are many programs that the user executes simultaneously which contains many processes that need the CPU to complete its task, but only one process can obtain the CPU at a certain time. Therefore, we need CPU scheduling, which helps to make the system more efficient and faster because it allows a process to use the CPU while another is on hold because it is waiting for other resources [1].

The CPU is one of the most important parts of the device. Most processes require it to be executed, so we need to maximize its utilization and throughput and minimize turnaround time, waiting time, and response time. All these criteria can be achieved using CPU scheduling algorithms that manage how processes enter the CPU [2].
There are many scheduling algorithms, and they are implemented in a different method. An example of these algorithms is the First Come First Served (FCFS) algorithm, which gives the CPU to the first arrived. Also, the Shortest Job First (SJF) algorithm, that gives the CPU to the short process. The Round Robin (RR) algorithm, which gives each process a time quantum, determines its working time in the CPU, and after the time expires, it exits the process and allows another to execute. The priority algorithm determines the entry of the process to the CPU based on their priority [1].

Many problems may occur during the scheduling algorithms' performance, and they may make it difficult to achieve the required criteria [3]. To the best of our knowledge, this paper reviewed the literature on the RR, FCFS, SJF, and Priority algorithms. To illustrate the methodology used to improve the performance of these algorithms, the results that were reached, several researchers have suggested various ways to improve CPU optimization criteria through different algorithms to improve the waiting time, response time, and turnaround time but there is no algorithm is better in all criteria.
This paper is structured as follows: Section 2 presents the literature selection methodology. Section 3 reviews the CPU scheduling algorithms. The RR scheduling algorithm is presented in section 4. SJF scheduling algorithm is mentioned in section 5. Section 6 reviews the FCFS scheduling algorithm. The priority algorithm is presented in section 7. Discussion and Conclusion are discussed in section 8 and section 9, respectively.

Manuscript received January 5, 2021
Manuscript revised January 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.1.4

2. Literature selection methodology

Literature selection methodology is the methodology that was followed in literature selection as follow:

2.1 A keyword Filter phase

The research articles were searched in the Google Scholar database using the following keywords: (1) Operating System (2) CPU management (3) CPU scheduling algorithms (4) RR scheduling algorithm (5) SJF scheduling algorithm (6) FCFS scheduling algorithm (7) Priority scheduling algorithm. The result of the research at this stage is 65 articles.

2.2 Abstract Filter phase

At this stage, the abstract is read to determine the research articles relevant to the research. The selection resulted in 40 articles from CPU scheduling algorithms, RR scheduling algorithm, SJF scheduling algorithm, FCFS scheduling algorithm, and Priority scheduling algorithm.
Due to the diversity of the CPU scheduling algorithm, many studies and research have improved and developed this algorithm to increase system performance. The following sections review several papers related to CPU scheduling algorithms where the studies have been categorized into four CPU scheduling algorithm types: RR, SJF, FCFS, and Priority Algorithms.

3. CPU scheduling algorithms

In this section, 4 articles on CPU scheduling algorithms, are reviewed and classified according to algorithms.
In [4], the researchers suggested comparing the three CPU algorithms based on each algorithm's waiting time to find the most appropriate algorithm for a particular process. They tested each algorithm individually and tested their results. In[5], the researchers proposed an improved version of the round-robin CPU scheduling algorithm based on the k-means clustering technique to combine the advantages of favor short process and low scheduling overhead of round-robin to reduce average waiting time, turnaround time. The k means algorithm is used to group similar processes in clusters. The proposed algorithm was compared with PWRR, TRR, PRR, SRR, and ADRR algorithms. The results showed that the proposed algorithm has a better performance by minimizing time cost compared with other algorithms.

In [6], these researchers created a fast system with fewer resources through the CPU scheduling algorithm. They reduced the algorithm's runtime and efficiency constraints. They implemented and developed algorithms FCFS, SJF, PS, RR, and DRR. Finally, they compared these algorithms, and DRR was the best among them. In [7], the researchers proposed scheduling algorithms to improve the operating system's real-time performance, and the CPU has been proposed. The proposed CPU is based on combining round scheduling (RR) and priority-based (PB) scheduling algorithms. Experimental results showed that the new algorithm improves all the round-robin scheduling algorithm CPU flaws.

4. RR scheduling algorithm

In this section, 18 articles on RR scheduling algorithms, are reviewed and classified according to algorithms. In [8], the researchers proposed a smart job first dynamic round-robin(SJFDRR) algorithm to solve the Round Robin scheduling algorithm problems related to time quantum. The algorithm is based on a dynamic-time-quantum approach used to make the CPU scheduler sort the process in ascending order on the burst time and assign the system priority and calculate a smart priority factor (SPF) for each process. The researchers designed a simulator to compare the proposed algorithm with the FJFDRR algorithm. The results showed that the proposed algorithm reduces the number of context switches, average waiting time, and average turnaround time when the arrival time of all process is zero. In [9], the researchers proposed an improved version of the Fittest Job First Dynamic Round Robin algorithm (FJFDRR) through added the process arrival time as an algorithmic factor implemented by multiple queues. The proposed algorithm was compared with FCFS, SJF, RR, and BJF algorithms in four test cases by used the ATAT, AWT, AR, and CS as metrics. The results showed that the proposed algorithm has the most balanced context switch degree based on the number of processes submitted. In [10], the researchers proposed Enhanced Round Robin (ERR) algorithm to improve CPU performance by minimizing the average waiting time and turnaround time. The proposed algorithm was compared with RR and IRR algorithms in three different cases. The results showed that the proposed algorithm has a better performance by reducing the average waiting time and average turnaround time compared with other algorithms.

In [11], the researchers are interested in time quantum problems in the round-robin algorithm. The Manhattan distance was used for the CPU burst times of processes to obtain the optimal time quantum value. ORRSM and SRR algorithms performance was compared in three cases with five processes that differed in burst times. The results showed that the ORRSM gives better performance of the RR algorithm with a reduction in context switches, turnaround times, and waiting times.

In [12], the researchers proposed a hybrid round-robin scheduling mechanism for process management (HYRR) to improve the RR algorithm. The HYRR algorithm inherits properties from the SJF and FCFS algorithms and dynamic time quantum was used instead of static time quantum. The HYRR algorithm was compared with the RR algorithm. The results showed that the HYRR algorithm has more effectiveness by maximizing CPU utilization, throughput and minimizing AWT, ATT, ART, and NOC.

In [13], the researchers proposed a load balancing policy between data centers and various load balancing solutions. In this paper, the researchers discussed the concept of pregnancy dispersion and then compared FCFS , SJF and RR on Cloudsim . The simulation results showed that RR task scheduling is much better than FCFS and SJF whether using time-sharing policy or shared space policy for cloudlet implementation.

In[14], The researchers applied the combined product between the SJF and RR algorithms to overcome the RR algorithm's flaws and reduce the response time and wait time in processes where the process is given a certain amount of time to execute. The results showed that the rated RR is good to use as it reduced latency and wait time and reduced starvation by implementing the least burst time first.

In [15], the researchers proposed scheduling a round-robin algorithm to obtain results and perform better from the round-robin (RR) algorithm. The proposed algorithm depends on determining the time quantum, which increases the round-robin (RR)scheduling algorithm's performance. The Experimental results The proposed algorithm more efficient through the identification of optimum time.

In [16], the researchers proposed a new CPU scheduling algorithm called Amended Dynamic Round Robin (ADRR) based on CPU burst time. It aims to improve the traditional RR scheduling algorithm using the concept of active quantitative time. The experimental results demonstrated that the proposed ADRR algorithm outperformed the other algorithms with less average waiting time, a small number of context switches, and less turnaround time.

In [17], The researchers proposed a new scheduling algorithm for the central processing unit called Efficient Round Robin Algorithm (ERRA) that differs from RR in that the time instead of being constant will be dynamic according to each process's needs to improve efficiency. The experimental results showed that the proposed algorithm gave better results in terms of average waiting time, average time spent, and changing the context. In [18], The researchers proposed a new approach for the RR algorithm, called the smart RR algorithm, which changes the time each time based on the remaining time of the process. The experimental results showed that this new approach from the RR algorithm improved lower average wait time and lower average response time than the traditional Round Robin scheduling method.

In [19], The researchers proposed a new algorithm, a set of algorithms that have been combined to produce an algorithm. Aiming to reduce the time spent by the process while it is waiting. It also reduces the number of switches to provide a good, fair, and effective algorithm. The results showed that the new algorithm increased productivity. In[20], The researchers proposed an improved algorithm from RR called DABRR (Dynamic Average Burst Round Robin). This algorithm uses the dynamic amount of time rather than the time used in the RR. The experimental results showed that DABRR better than RR. In [21], have been suggested a new CPU scheduling algorithm has been proposed to incorporate a round-robin and priority scheduling algorithm. That provided a solution to the problem of aging by prioritizing operations and working to reduce famine. The experimental results showed that the proposed algorithm overcomes all the drawbacks of the existing round-robin algorithm.

In [22], have been suggested a new CPU scheduling algorithm has been proposed, which is an improved version of the Round Robin algorithm that relies on intelligently calculating quantitative time and prioritizing operations based on it. The experimental results showed that the proposed algorithm outperformed from where less average waiting time, a small number of context switches, and less turnaround time than the simple Round Robin algorithm. In [23], The researchers proposed a dynamic time quantum-based Round Robin (DTQRR) algorithm to improve the CPU performance by using features of IRR and IRRVQ to reduce the waiting time, turnaround time, and the number of context switches. The proposed algorithm was compared with RR, IRR, and IRRVQ algorithms on waiting time, turnaround time, and context switch. The results show that the performance of proposed algorithms is better than the RR, IRR, and IRRVQ as it reduces all three scheduling criteria.

In [24], the researchers developed a simplified dynamic improved round-robin (STARR) CPU scheduling algorithm to enhance the round-robin algorithm. The proposed algorithm depends on the numeric outlier detection technique and geometric mean to determine an optimal time quantum when the burst times of processes arriving in the ready queue include outliers. The performance of the proposed algorithm was compared with selected improved versions of RR. The result showed that the proposed algorithm performed better in reducing the average waiting time and average turnaround time. In [25], the researchers proposed a new algorithm to improve the CPU's performance, derived from the features provided by the Round Robin algorithm. The results showed that the proposed algorithm offers better than the Round Robin in terms of average waiting time, average turnaround time, and several context switches.

5. SJF scheduling algorithm

In this section, 13 articles on SJF scheduling algorithms, are reviewed and classified according to algorithms. In [26], the researchers introduced a comparison between FCFS and SJF to show which algorithm is the process that is more suitable and how their scheduling is being implemented. They Performed through the use of exemplary job processes to determine the best among the algorithms. The result showed the SJF gives better performance for scheduling processes than the FCFS.

In [27], the researchers suggested a comparison between FCFS and SJF for scheduling a multimedia operation and analyzing the variance in their performance through different sets of data and a variety of operations. The results showed that SJF for multimedia task scheduling performs better than FCFS after applying it to three different data groups. In [28], the researchers proposed a comparison between the Round Robin scheduling algorithm and the shortest job first scheduling algorithm and the proposed algorithm. There were different scheduler levels applied at different levels of the process, from the ready queue to terminate. The results showed that the proposed scheduling had less waiting time, response time, context switching, and less preventive than the RR scheduling algorithm and less wait time than the SJF scheduling algorithm. In [29], researchers analyzed an FCFS and SJF-based priority scheduling algorithm for similar priority jobs to decide which processes in the ready queue will be assigned to the CPU. The experimental result shows that the average waiting time and average turnaround time are reduced. In [30] the researcher proposed a system that works on the SJF algorithm. The operation is carried out according to the time of the explosion. Initially, processes are assigned to the ready-made queue according to priority. The results showed that the algorithm running on SJF is better than FCFS and has a better average wait time.

In [31], The researchers applied the Least Slack Time First and SJF algorithm for the Realtime system to determine the dynamic priority according to the slack time. The task that has a minimum slack time has the highest priority. The results showed that the LST algorithm works fine at low load and bad in an overload condition. As for SJF, it cannot schedule a job at a low load while it performs relatively well in case of overload. In [32], Researchers have proposed K FACTOR algorithm to reduce SJF's limitations and overcome its shortcomings (starvation and polarization towards shorter operations). A parameter called K is used, which is initialized with values and increases with each node. The results showed that the proposed algorithm is not better than SJF, but it eliminates starvation for long operations to become fair for all operations.

In [33], have discussed the scheduling algorithms of the different CPUs and clarified the best algorithms for the special case in terms of waiting time and response time compatible with scheduling goals. The Experimental results have shown that the Shortest Job First Scheduling (SJF) tends to increase wait time, making presenting long processes impossible. In [34], have discussed the scheduling algorithms used to schedule operations in the CPU multi-programming system, first come first served (FCFS), round-robin (RR), shortest job first (SJF), made a comparison was based on scheduling criteria eight (8) Important parameters in process scheduling. The experimental results showed that the ideal proposed algorithm should reduce response time and overheads in terms of CPU, disk, and memory usage and increase productivity.

In [35], have discussed the importance of CPU scheduling by switching between different processes to make the most of the CPU and clarify the best algorithm for a special case in terms of waiting time and response. The Experimental results have shown that the SJF scheduling algorithm gives the minimum average standby time. In[36], Researchers have proposed improving the algorithm SJF in cloud computing. By releasing a new algorithm, a modified Shortest Task First (MSJF) algorithm to reduce Makespan and reduce average response time while maximizing resource usage. The experimental results showed that MSJF is better than SJF and FCFS.

In [37], The researcher suggested an ideal algorithm for SJF in which adjustments in the shortest time would be chosen. In this algorithm, processes with the shortest time to complete are placed in front of the ready queue. Results showed that SJF is suitable for batch jobs where the runtime is predefined, and each process can be executed based on minimum burst time. In [38], the researchers presented a new algorithm SJF approach to improve CPU efficiency. A comparative analysis of the proposed algorithm was performed using Round-robin algorithms and SJF protective algorithms. The results showed that the proposed algorithm improved the system's performance by reducing the context switching to the required extent.

6. FCFS scheduling algorithm

In this section, 2 articles on FCFS scheduling algorithms, are reviewed and classified according to algorithms. In [39], the researchers proposed comparing SJF and FCFS for best utilization of memory. The results showed SJF best in less average waiting time between multiple scheduling algorithms and FCFS best in a simple application, which does not involve any complex logic.

In[40], the researchers designed the MapReduce Analysis Process Management System to perform the MapReduce function in a process management application to use the BPM (BigData Process Management) engine by discussing a tool that fits the functionality by analyzing the characteristics and performance of the models. FCFS is the most suitable data analysis algorithm, being the effective implementation of data set and data.

7. Priority scheduling algorithm

In this section, 3 articles on Priority scheduling algorithms, are reviewed and classified according to algorithms. In [41], the researchers proposed a new CPU scheduling algorithm called the Modified Priority Preemptive Scheduling Algorithm. The algorithm circularly implements priority pre-scheduling. The results showed that the new solution solved the starvation problem and enhanced the normal Preemptive algorithm performance.

In [42], researchers proposed an algorithm for load balancing and prioritizing applications of prime importance. The SJF algorithm has to be used for ranking jobs and then the RR algorithm must be used for processing. The results showed that the ad hoc algorithm pays attention to higher priority tasks and executes them quickly and for low priority operations, reduces contextual switching and hence the choice between RR and SJF.

In [43], The researcher proposed a scheduling algorithm for high priority Cyber-Physical System random tasks. A fog group is used to process the last available time and execution time and then give the system to the random task with high priority in idle time. The algorithm improves the sending rate of random tasks that have high priority and can be executed more efficiently.

8. Discussion

CPU scheduling algorithms have helped to manage and schedule the process in the operating systems. After reviewing studies from [4] to [7], based on this generally. The results showed each algorithm has the characteristics that distinguish it based on a certain criterion. The percentages of results are shown in Figure1. The largest percentage of 50% is reached in the RR algorithm. Then with 25% in SJF. Finally, a percentage of 25% in FCFS.

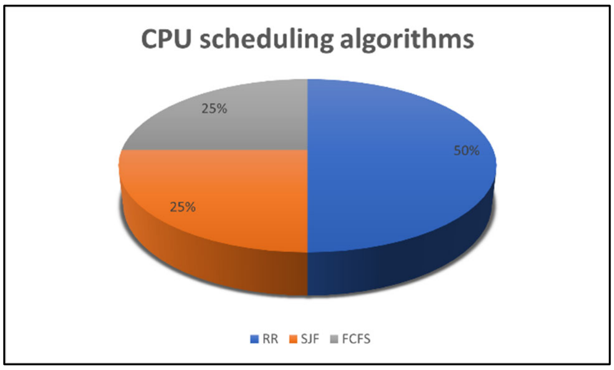

Fig. 1. The percentage of studies on the CPU scheduling algorithm

After reviewing studies from [8] to [25] based on RR. The results showed that the RR algorithm improved the average waiting time, throughput time, turnaround time, and response time. It is also maximizing CPU utilization. The percentages of results are shown in Figure2. The largest percentage of 68% is reached in the RR algorithm only. Then with 16% in SJF with RR. Then with 4% in Priority with RR. Finally, a percentage of 12% in FCFS with RR.

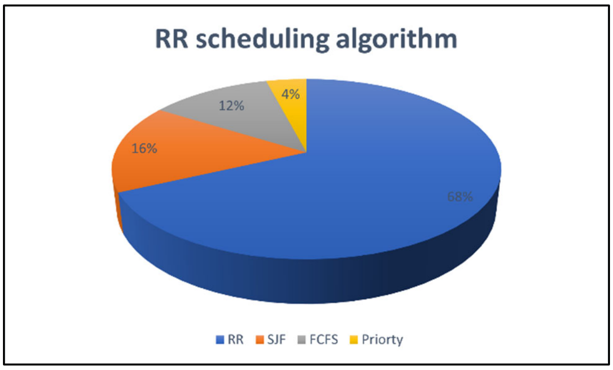

Fig. 2. The percentage of studies on the RR scheduling algorithm

After reviewing studies from [26] to [38] based on the SJF scheduling algorithm. The results showed that the SJF algorithm reduced the average waiting time, turnaround time, and response time. The percentages of results are shown in Figure3. The largest percentage of 56% is reached in the SJF algorithm. Then with 26% in merging FCFS with SJF. Finally, a percentage of 9% in merging RR and FCFS with SJF.

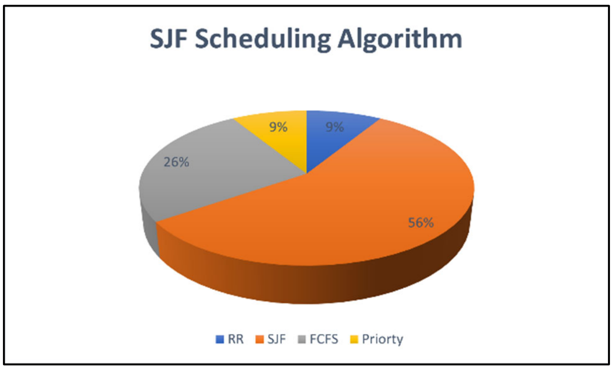

Fig. 3. The percentage of studies on SJF scheduling algorithm

After reviewing studies from [39] to [40] based on the FCFS scheduling algorithm. It is best to reduce the average waiting time. The FCFS algorithm is most suitable for data analysis operations and the best simple to apply. The percentages of results are shown in Figure4. The largest percentage of 67% is reached in the FCFS algorithm. Finally, a percentage of 33% in SJF with FCFS.

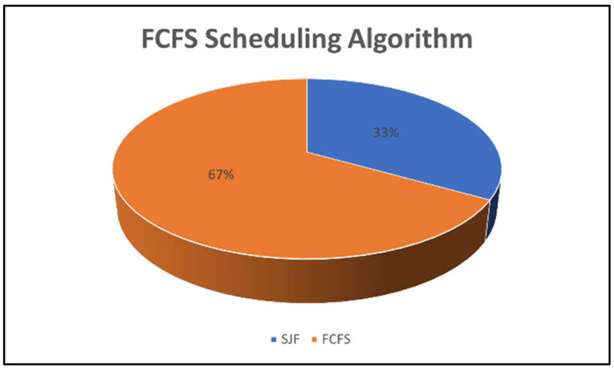

Fig. 4. The percentage of studies on the FCFS scheduling algorithm

After reviewing studies from [41] to [43] based on the priority algorithm. The results showed that the Priority algorithm is concerned with the highest priority operations and rapid implementation, solving the starvation problem. The percentages of results are shown in Figure5. The largest percentage of 60% is reached in the Priority algorithm. Finally, a percentage of 20% in RR and SJF merged with the priority algorithm.

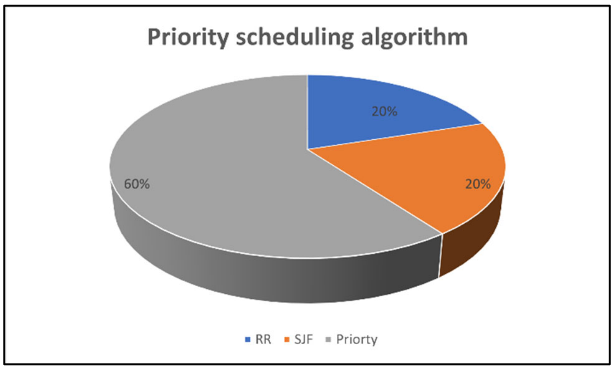

Fig. 5. The percentage of studies on the Priority scheduling algorithm

9. Conclusion

This paper presented studies related to CPU scheduling algorithms. This study showed that optimized scheduling algorithms for different CPUs could reduce response time, wait time, and overheads in CPU, disk, and memory usage and increase productivity. In general, we believe that optimized CPU algorithm scheduling improves CPU performance, just as the results showed that CPU algorithm scheduling helps reduce wait time and response time for processes. Also, the CPU algorithms allow the user to get good results without wasting valuable time. In the future, other researchers can conduct more research and investigations to explore other scheduling algorithms that will be optimal and thus provide optimum user satisfaction.

References

[1] Galvin, P. B., Gagne, G., & Silberschatz, A. (2003). Operating system concepts. John Wiley & Sons.

[2] Abbas, A. (2015). Multiprocessor and Real-Time Scheduling Shortest-Job-First (SJF) Scheduling Algorithm. European Journal of Computer Science and Information Technology, 3(5), 8–20.

[3] Kamalam, G. K., & Bhaskaran, D. V. M. (2014). Resource Fitness Task Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment. Australian Journal of Basic and Applied Sciences, 8(18), 128-135.

[4] Siahaan, A. P. U. (2016). Comparison analysis of CPU scheduling: FCFS, SJF and Round Robin. International Journal of Engineering Development and Research, 4(3), 124-132.

[5] M Mostafa, S., & Amano, H. (2020). Dynamic Round Robin CPU Scheduling Algorithm Based on K-Means Clustering Technique. Applied Sciences, 10(15), 5134.

[6] Farooq, M. U., Shakoor, A., & Siddique, A. B. (2017, March). An efficient dynamic round robin algorithm for cpu scheduling. In 2017 International Conference on Communication, Computing and Digital Systems (C-CODE) (pp. 244-248). IEEE.

[7] Zouaoui, S., Boussaid, L., & Mtibaa, A. (2019). Priority based round robin (PBRR) CPU scheduling algorithm. International Journal of Electrical & Computer Engineering (2088-8708), 9(1).Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 397-400). IEEE.

[8] Gupta, A. K., Yadav, N. S., & Goyal, D. (2016). Design and Performance Evaluation of Smart Job First Dynamic Round Robin (SJFDRR) Scheduling Algorithm with Smart Time Quantum. American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), 26(4), 66-78.

[9] Manuel, J. I., Baquirin, R. B., Guevara, K. S., & Tandingan, D. (2019, February). Fittest Job First Dynamic Round Robin (FJFDRR) scheduling algorithm using dual queue and arrival time factor: a comparison. In IOP Publishing Ltd, IOP Conf. Ser.: Mater. Sci. Eng (Vol. 482, p. 012046).

[10] Khatri, J. (2016). An enhanced Round Robin CPU scheduling algorithm. IOSR Journal of Computer Engineering (IOSR-JCE), 18(4), 20-24.

[11] Srilatha, N., Sravani, M., & Divya, Y. (2017). Optimal Round Robin CPU Scheduling Algorithm Using Manhattan Distance. International Journal of Electrical and Computer Engineering, 7(6), 3664

[12] Ali, K. F., Marikal, A., & Kumar, K. A. A Hybrid Round Robin Scheduling Mechanism for Process Management.

International Journal of Computer Applications, 975, 8887.

[13] Pratap, R., & Zaidi, T. (2018, August). Comparative study of task scheduling algorithms through Cloudsim. In 2018 7th International Conference on Reliability, Infocom

[14] Srujana, R., Roopa, Y. M., & Mohan, M. D. S. K. (2019, April). Sorted Round Robin Algorithm. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 968-971). IEEE.

[15] Biswas, D., & Samsuddoha, M. (2019). Determining Proficient Time Quantum to Improve the Performance of Round Robin Scheduling Algorithm. International Journal of Modern Education and Computer Science, 11(10), 33.

[16] Shafi, U., Shah, M. A., Wahid, A., Abbasi, K., Javaid, Q., Asghar, M., & Haider, M. (2020). A novel amended dynamic round robin scheduling algorithm for timeshared systems. Int. Arab J. Inf. Technol., 17(1), 90-98.

[17] Aijaz, M., Tariq, R., Ghori, M., Rizvi, S. W., & Qazi, E. F. (2019, March). Efficient Round Robin Algorithm (ERRA) using the Average Burst Time. In 2019 International Conference on Information Science and Communication Technology (ICISCT) (pp. 1-5). IEEE.

[18] Mody, S., & Mirkar, S. (2019, December). Smart Round Robin CPU Scheduling Algorithm For Operating Systems. In 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEEECCOT) (pp. 309-316). IEEE.

[19] Parekh, H. B., & Chaudhari, S. (2016, December). Improved Round Robin CPU scheduling algorithm: Round Robin, Shortest Job First and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time. In 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC) (pp. 184-187). IEEE.

[20] Alworafi, M. A., Dhari, A., Al-Hashmi, A. A., & Darem, A. B. (2016, December). An improved SJF scheduling algorithm in cloud computing environment. In 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEEECCOT) (pp. 208-212). IEEE.

[21] Joshi, A., & Gosswami, S. (2017). Modified Round Robin algorithm by using Priority Scheduling. Advances in Computational Sciences and technology, 10(6), 1543-1549.

[22] Zouaoui, S., Boussaid, L., & Mtibaa, A. (2019). Priority based round robin (PBRR) CPU scheduling algorithm. International Journal of Electrical & Computer Engineering (2088-8708), 9(1).

[23] Berhanu, Y., Alemu, A., & Mishra, M. K. (2017). Dynamic time quantum based round robin CPU scheduling algorithm (Doctoral dissertation).

[24] Omotehinwa, T. O., Azeez, I., & Oyekanmi, E. O. (2019). An Improved Round Robin CPU Scheduling Algorithm for Asymmetrically Distributed Burst Times.

[25] Berhanu, Y., Alemu, A., & Mishra, M. K. (2017). Dynamic time quantum based round robin CPU scheduling algorithm (Doctoral dissertation).

[26] Bibu, G. D., & Nwankwo, G. C. (2019). COMPARATIVE ANALYSIS BETWEEN FIRST-COME-FIRST-SERVE (FCFS) AND SHORTEST-JOB-FIRST (SJF) SCHEDULING ALGORITHMS.

[27] Magdalene, R., & Sridharan, D. (2020). Comparative Analysis of FCFS and SJF for Multimedia Process Scheduling. In Advances in Communication Systems and Networks (pp. 639-646). Springer, Singapore.

[28] Kumar, S., Kumar, G., Jain, K., & Jain, A. (2018). An approach to reduce turn around time and waiting time by the selection of round robin and shortest job first algorithm. International Journal of Engineering & Technology, 7(2.8), 667-672.

[29] Chandra Shekar, N., & Karthik, V. (2017). Analysis of Priority Scheduling Algorithm on the Basis of FCFS & SJF for Similar Priority Jobs. International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), 4(3).

[30] Chauhan, H., & Inani, A. (2019, June). Modified Concept to Achieve Maximum Efficiency of CPU Scheduling Algorithm. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 660-663). IEEE.

[31] Teraiya, J., & Shah, A. (2018, September). Comparative Study of LST and SJF Scheduling Algorithm in Soft Real-Time System with its Implementation and Analysis. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 706-711). IEEE.

[32] Garg, S., & Kumar, D. (2018, July). A k-Factor CPU Scheduling Algorithm. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[33] Goel, N., & Garg, R. B. (2013). A comparative study of cpu scheduling algorithms. arXiv preprint arXiv:1307.4165.

[34] Adekunle, Y. A., Ogunwobi, Z. O., Jerry, A. S., Efuwape, B. T., Ebiesuwa, S., & Ainam, J. P. (2014). A comparative study of scheduling algorithms for multiprogramming in real-time systems. International Journal of Innovation and Scientific Research, 12(1), 180-185.

[35] Pushpraj, S., Vinod, S., & Anjani, P. (2014). Analysis and Comparison of CPU Scheduling Algorithms. International Journal of Research in Engineering & Technology (IJRET) Vol, 4(1).

[36] Dash, A. R., & Samantra, S. K. (2016). An optimized round Robin CPU scheduling algorithm with dynamic time quantum. arXiv preprint arXiv:1605.00362.

[37] Putra, T. D. Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling.

[38] Hamayun, M., & Khurshid, H. (2015). An optimized shortest job first scheduling algorithm for CPU scheduling. J. Appl. Environ. Biol. Sci, 5(12), 42-46.

[39] Sowmya, G., ChinaAppalaNaidu, R., Meghana, K., & Bandi, R. (2018). A COMPARISON OF

SCHEDULING ALGORITHM FOR BEST UTILIZATION OF MEMORY. International Journal of Pure and Applied Mathematics, 120(6), 3563-3570.

[40] Krishna, M. V. (2018). BigData Processing using First Come First Served (FCFS) Algorithm. International Journal of Computer Science and Mobile Computing, 7(7), 83-87.

[41] Chandiramani, K., Verma, R., & Sivagami, M. (2019). A Modified Priority Preemptive Algorithm for CPU Scheduling. Procedia Computer Science, 165, 363-369.

[42] Tripathi, S., Prajapati, S., & Ansari, N. A. (2017, May). Modified optimal algorithm: for load balancing in cloud computing. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 116-121). IEEE.

[43] Zhang, J., Chen, C., Zheng, H. K., & Luo, Q. Y. (2019, June). A High Priority Random Task Fuzzy Scheduling Algorithm for CPS. In 2019 Chinese Control And Decision Conference (CCDC) (pp. 482-487). IEEE.