

CPU Scheduling Techniques: A Review on Novel Approaches

Strategy and Performance Assessment

Abstract

The operating system (OS) is now being widely used in many of the various modern applications in recent years, ranging from diabetic monitoring to other strategic planning. The main function of any OS is to control and coordinate all hardware resources among the commands of the same/different user, which increases the efficiency of advanced comprehensive applications. In an autonomous computer system, the CPU is one of the important resources to manage and process all activities which require scheduling techniques on a processor. Since the early days of computing and other multi-programming OS, various studies have been assigned to CPU scheduling techniques based on processes management and performance evaluation. Thus, outlining the many issues related to scheduling methodologies and the weaknesses that need to be addressed. This review paper is organized based on two distinct perspectives: the implement strategies of CPU scheduling technique and criteria-based measures used, which assess how the strategies are analyzed and used under performance evaluation.

Keywords: Operating system, CPU scheduling techniques, implemented strategy, performance assessment.

I. INTRODUCTION

All modern devices today operate within software systems. The operating system (OS) is a software system that is installed in computers and other devices to provide a suitable environment for each user to implement their programs in an advanced manner. The OS has many important goals including user deployment, operations, reliability, performance, and control [1][2]. Multiprogramming and multitasking OSs are the main reasons why modern computer systems need CPU scheduling. The relatively newly released advanced OS has specific CPU scheduling, which also requires many efficient processes to use the central processing unit at simultaneous times.[3][4]. The scheduler is special system software that is responsible for handling resource scheduling from the queue services resource. The scheduler can be divided into three major categories. The first category is a long-scheduler, which defines the process that leads to the Random Access Memory (RAM) from the process pool hard-disk (HDD). The second category is called a short term scheduler, which decides which process to be executed on the CPU selected from the ready queue (RQ). The

third type is called medium scheduling, which alters the process to reduce the degree of multiprogramming and the CPU overhead.[5][6]. Fig.1 shows the different schedulers (timelines).

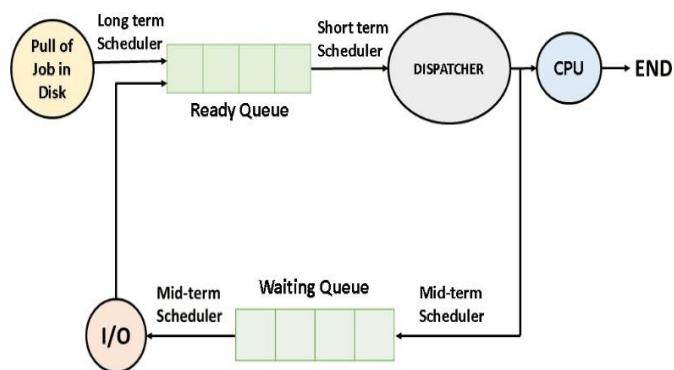


Fig. 1. Diagram of Process Schedulers

CPU scheduling management techniques are intended to decide on short-term scheduling of all ready execution processes. CPU scheduling approaches are the core functions of a multi-programming OS which is tested to maximize CPU use[7][8].

To develop effective scheduling techniques, we need to understand clearly the issues related to the different scheduling methodologies and the drawbacks which need to be overcome. This paper aims to provide a detailed review of modern work in scheduling techniques and identify metrics that are suitable for CPU scheduling performance.

This paper is organized into five different sections. The first one is a general introduction. The second section illustrates the development of various CPU scheduling techniques. Section three presents a literature review based on techniques and the evolution of performance. The discussion of classifications, contribution, and analysis of the solution methods is summarized in section four. The final section provides some conclusions and future work in section five.

II. DEVELOPMENT OF CPU SCHEDULING TECHNIQUES

Scheduling system algorithms come in two basic types: (1) non-preemption, which means that, once the process is running it will continue to run until the CPU is out of control, while (2) preemption involves interrupting the process that is currently running for a certain critical CPU command [9].

There are several required criteria measures for scheduling: maximizing throughput, overhead CPU, minimizing turnaround, processing time, and response time. The last criteria measure is impartially handling long-term processes and ensuring that light processes can be addressed quickly[10][11]. Fig. 2 indicates the major CPU scheduling techniques to make short scheduling decisions among several ready processes[12].

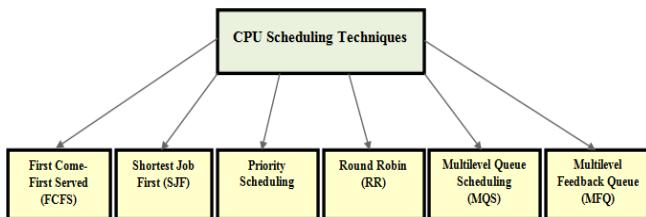


Fig. 2. Common CPU Scheduling Techniques

The first category is the simplest programming discipline known as First Come First Serve (FCFS). This technique's philosophy is based on the process's arrival to the queue of processes. In other words, the first process assigned to the processor, and this type is called a non-preemptive scheduler. Shortest Job First (SJF) category is a common discipline of preferences. The method has the smallest requirements for the processing which assigns processes to the CPU first. This type of scheduling can be non-preemptive and preemptive, and it can also be used in batch systems. A special CPU algorithm is Priority scheduling which is the third category this technique's characteristic may be preemptive scheduling, and the process may also evaluate internal or external. The major issue with priority scheduling is indefinite blocking (starvation), the method used to resolve this problem termed as ageing. The

fourth category is a round-robin (RR) every process is set to a quantity of time in execution. If it is not done within the quantum time, then the processor is allocated to a new process, and the old process should be managed and returned to the waiting queues. The fifth category of scheduling techniques is the Multilevel Queue Scheduling (MQS) the property of this algorithm is dividing the ready queues into several separate queues. The final category is a Multilevel-Feedback Queue (MFQ) (sometimes called a "front-ground/back-ground multilevel"). The purpose of this strategy is to split the ready queue into various queues and the cycle will proceed through several separate queues; the ageing technique can be implemented in this way so that each queue is given a particular priority level, and each queue follows the scheduling [13][14].

III. LITERATURE REVIEWS

In this section, all the articles considered in the review are classified based on different criteria: approach strategy and criteria-based on measures. Some of the proposed scheduling techniques are outlined briefly.

Fawad Ahmad et al. in [16], Proposed hybrid scheduling discipline more effective and improves the lack in the current scheduling techniques. It assists users to utilize, and study the multi-programming environment in an OS. Moreover, Robert I. Davis & Marko Bertogna in [17] Prepare an interfaced to simulate the microscopic traffic, with an AVR 32-bit microcontroller, and accomplish the real time (RT) estimate of four scheduling techniques. Jian-Jia Chen et al in [18], Introduced a "fixed-priority scheduling strategy" optimization approach with deferred preemption on a uniprocessor. This strategy is guaranteed to find a scheduled priority, and optimal method depends on determining the minimum length of Final Non-preemption Region and Priority Assignment (FNR-PA) for each task and also ensures work on the multi-processor case. Also, Munam Ali Shah et al in [19] Designed a novel priority task polices as a result obtained the identical scalability tests, furthermore, managing the importance of mixed hard-real-time and soft-real-time separate tasks in the system. Abbas Noon et al in [20] Proposed a novel scheduling strategy called "intelligent method scheduling algorithm based on an entity" (OIPSA). OIPSA decreases all response-time, wait-time and turnaround time for desired processes in the strategy and increases overall system efficiency when dealing with conventional scheduling techniques.

Ahmed Alsheikhy et al in [21], Presented a solution to the time slice issues by making an OS regulate the time slice based on dynamic the execution time on the existed processes in the RQ. Shihab Ullah in [22] Developed a novel enhanced dynamic Round-Robin (RR) technique which contributes either on a single processor or multi-processor systems. The major goals of this technique are to reduce the average of waiting-time and turnaround time while maximizing throughput over the overhead resolving from context-switching between various tasks. Moreover, Liang Luo et al in [23] modified "optimum dynamic time quantum using round-robin" scheduling (ODTQRR). Further than the use of the several-queues form of existing scheduling, also used a synchronous dynamic queue as either blocking or ready queue respectively. Moreover, Liang Luo et al [24] Proposed method called a resource scheduling

technique of cloud computing based on optimum energy efficiency. Firstly, studying the energy-wasting model, and order the computing resource in four main parts: main memory, CPU, network, and auxiliary storage. Finally simulate the consequence and analyses demonstrate the scheduling performance. Priyanka Sangwan et al in [25], Comparing the original RR with a proposed model called a traditional resource scheduling algorithm of cloud computing to enhance balance the loading in the cloud environment and displayed the results.

Prakash Patil & Chaitali Borse in [26], Proposed an opportunistic RR scheduling technique, that the multiple users have the chance to send the data and information. It is both easy and simple for the accomplishment. In this scheduling algorithm, only multiple users can receive and transmit, and thus it doesn't warranty the fairness in resource allocate to "multiple inputs multiple outputs" MIMO wireless system. Vaishali Chahar & Supriya Raheja in [27], Proposed a novel fuzzy-based on multi-level queue CPU scheduling scheme. In the proposed technique, the CPU service time is allocated and the value of time-quantum (TQ) calculates the values dynamically among the different queues. The scheduling technique implemented using the "fuzzy toolbox" provide by the Math-lab program. Similarly, Ali Rezaee et al [28] Proposed the "fuzzy dynamic based weighted fair queue" (FDWFQ) to take place flow in splits queues where weight is used to define how many packets are managed at a time. Defining weights relies on the serving requirement and queue weight are allocated dynamically using the feedback of quality of serving and a fuzzy inference system metrics. Supriya Raheja et al [29], Introducing an approach to contribute towards the enhancement of the MLQ approach, that is, "vague logic". VL is an extended form of fuzzy-logic, which become a dedicated tool to handle vague information. Also, it focuses on enhancing the performance of MLQ scheduling technique and resolves the indefinite blocking problem of lower priority process.

On the other hand, Saiyed Makhduma F et al [30] Presents multilevel queue model for "Indian region broadcast network" (IRBN) through satellite. In this model the data are transmitted on a priority task basis, so more queues are required, which stream all data that are to be dynamically telecast. Seifiedine Kadry & Armen Bagdasaryan in [31], Presented the novel MLFQ that was efficient in dealing with lower priority jobs (CPU-Bounded) and convenient with higher priority jobs (I/O-Bounded), it solves the problems of response-time. So the simulation of the MLFQ scheduling technique doing best with compared to the classical MLFQ using static and dynamic quantum. Malhar Thombare et al in [32] Proposed an easier method to generation automatic slice time to reduce the starvation, prioritizing of several types of processes equally at the same period time aspect if each process can effectively get a fair opportunity to get CPU service time, and resources to get running. Moreover, Sanjay. K. Dwivedi and Ritesh Gupta [33] Developed an approach utilizes the improved method of dynamic quantum time generation to reduce the waiting- time. MFQ improves by implementing SJF selection before the RR scheduling technique from the 2nd queue onwards (next to SJF-RR) using automatically generated time slice as compared to other scheduling techniques and for static time quantum. Kenneth E. Hoganson in [34], Investigate the researches of

MLFQ for "real-time" OS scheduling, incorporating a novel method and technique for mitigating the indefinite blocking problem of MLFQ, which periodically follow up time from intermediate queues to the lower priority queues. Moreover, Spencer Killen et al [35], presented a Visualizations Gantt-charts for the scheduling technique of "Multi-Level Feedback-Queues," to evaluate the architectural design and essential features of the visualization software, and to use the method in educational environments. Additionally, provide AV tools in order to provide visuals decoding the existing state of the simulation processes.

Shweta Jain & Saurabh Jain [36] offered simulation study called Markov-chain analyzes for the scheduling technique of "multi-level feedback-queue," which detects an effect of the waiting state with total system efficiency and throughput. This research also underlines the estimation of the comparative analysis using varying α and d values in an arithmetic model.

Anita Path & Rajashree.V.Biradar in [37], Reviewed Tiny OS based on the requirement of the application, and scheduling strategy. Here, features of the Tiny OS are summarized out and other different scheduling schemes, proposed by many researchers are calculated. Many of the authors used power TOSIM and TOSSIM as simulators to confirm the results. Next, Oshin & Amit Chhabra in [38] Provides a comparative analysis of various ant-colony optimization "ACO's" variants according to their performance in defining near-optimal solutions. The consequence is drawn from the Investigate ant-colony optimization efficiency in resolving the different scheduling scheme problems. Similarly, Tabassum A. Maktum et al in [39] Proposed idea to find near-optimal solutions by implementing "Genetic algorithm" for the CPU scheduling technique problem. They developed a simple scheduling algorithm according to genetic-approach for scheduling processes on uniprocessor and have compared reduce average waiting-time of their scheduling algorithm with SJF and FCFS scheduling. Moreover, Shatha Jawad in [40] Proposed Neuro-fuzzy-based scheduling technique in the applying and modified CPU scheduling algorithm to maximize the time of response and reduce average waiting time, and turn-around time. This is achieved by combining well-known scheduling schemes, including pre-emptive SJF and pre-emptive priority scheduling with the Neuro-fuzzy method.

On the other hand, Tarek Helmy et al in [41], Suggested a machine-learning method that can be implemented in "Computational-Grids" (CGs) to predict the exact timing of the CPU burst for the processes to be recognized. The proposed strategy benefits from analyzing the most important features for a machine, so it can predict the next period of execution of the CPU carefully. Moreover, Umadevi. K.S et al [42] Implement the software defined network (SDN) using an open-flow protocol, which is confirmed by setting protocols of communication and APIs. The multi-level scheduling technique picket the priority role of SDN of switches, which impact the effective management of data traffic. Sushil Kumar Saroj et al in [43], The proposed technique is simulated and implemented in C++ programming. This approach, introduces other issues such as long-average waiting times, turnaround times, indefinite blocking (starvation) and practical implementation. The suggested scheduling has removed the many problems and also

simple to perform by employing two types of slice time. In addition, Jinkyu Lee et al in [44] Proposed control pre-emption (cP-EDF), which controls, the cases of pre-empting processes, while current approaches strategies arrange that of pre-empted processes. This technique also focused on a single processor platform, which can provide at most one process at a time. Yosef Hasan Jbara [45], Developed an educational simulation tool using a visual basic application (VBA), to visually demonstrate various scheduling techniques for uniprocessor. The developed tool can be used for realizing learning and teaching goals. This work display in details the various features and advantage of the sophisticated tool. Omar Ahmed & Adnan Brifcani in [46], Applied deep learning utilizes a multi-layered artificial neural network, such as through as deep neural, and optimal neural networks for learning advanced features deep learning uses the essential features learned from data.

IV. CLASSIFICATION AND DISCUSSION

A. Classification based on scheduling Techniques

Many existing research papers done in the many fields of scheduling techniques in the CPU are discussed briefly. A huge

number of researches have been devoted to fixed priority strategies, quantum time and other approaches to work through drawback in scheduling algorithm. This section reviews and categorizes the different CPU scheduling approaches. They can be classified into seven broad scheduling techniques which listed in Table I: FCFS, SJF preemptive and non-preemptive, preemptive priority, non-preemptive priority, RR, MLQS, MLFQS, and other scheduling based on approach strategy [15].

Table I indicates when each of the scheduling techniques has been implemented and the research paper that has been used for various years. Here, we divided the current task scheduling study into three separate periods as follows: 2011-2013, 2014-2016, and 2017-2019. The recent research works have been classified by the methodology used. This analysis would be very useful in understanding which strategies have been commonly used in earlier days and how newer algorithms have been implemented based on additional requirements.

TABLE I. STATE OF ART ANALYSIS OF REVIEW FROM 2011-TO-2019

CPU Scheduling	Article Durations from 2011-to- 2019		
	2011 - 2013	2014 - 2016	2017 - 2019
FCFS (Non- preemptive Scheduling)		✓	
		✓	
SJF (Non- Preemptive and SJF Preemptive Scheduling)		✓	
	✓		
Priority (Non- pre-emptive and pre-emptive Scheduling)	✓		
		✓	✓
RR (pre-emptive Scheduling)	✓		
		✓	
MLQ (Multilevel Queuing Scheduling)			✓
		✓	✓
MFQS (Multilevel Feedback Queue Scheduling)			✓
		✓	✓
Other Scheduling Techniques	✓		
		✓	
		✓	
		✓	
		✓	
		✓	

B. Classification based on scheduling Techniques

In this classification, the scheduling technique mentioned in each of the 12 research papers considered is identified. Table II.

Clearly outlines many of the latest implementation practices in scheduling schemes worked in different strategies. The following table summarizes the reviewed scheduling techniques, and it's a strategy implemented in brief.

TABLE II. SUMMARIZES THE REVIEWED PROPOSED SCHEDULING STRATEGIES

Ref. No	New Proposed Scheduling Technique	The Implementation Strategy
[16]	Hybrid discipline scheduling SJF with RR Scheduling.	Hybrid discipline scheduling, more effectively and enhance the lack in the current scheduling techniques. Also, support the multi-programming environment.
[17]	Simulate the microscopic traffic congestion control based schedulers real-time.	Implemented an interfaced to simulate the microscopic traffic congestion control, it's more benefited for collecting vehicle information.
[18]	Optimizing Fixed Priority Pre-emption Scheduling Technique.	Set a fixed priority scheduling with the technique of deferred pre-emption (FP-DS) is to find a scheduled priority, and the optimal solution depends on finding the minimum final, non-preemptive region (FNR).
[19]	New Fixed-Priority Pre-emption Scheduling Technique.	Mixed-soft and hard real-time to support the identical scalability tests on multi-processors.
[24]	A Resource Scheduling Technique of Cloud Computing.	Supports energy-efficient optimization technique, safely decrease energy consumption.
[25]	Round Robin (RR) Scheduling in cloud computing.	Realize a superior work and enhance balance the load in the cloud computing environment.
[26]	Opportunistic Round Robin Scheduling Technique	Multiple users have the opportunity to receive and transmit data and equip fairness to all users to allocate resources.
[27]	Fuzzy based multilevel queue CPU scheduling Technique.	Implemented fuzzy multilevel queue scheduling by using the Math-lab program to compute the values dynamically among the different queues.
[29]	Vague Logic Based (MLQ) Scheduling Technique.	Based on Vague Logic, The (MLQ) prevent the lower priority process from the indefinite blocking problem.
[35]	An Educational Visualization Tool Based on (MLFQ) Scheduling Technique.	The new tools support a set of classes which learn important MLFQ scheduling concepts by providing a new Gantt-chart for the current state of processes.
[42]	Software-Defined Networks Based on (MLFQ) Scheduling Technique.	Implement software-defined networks used in the open flow protocol. The echoes used an OMNeT++ simulator, it shows an effective manner of processing the incoming data traffic.
[44]	Preemption Policy of cP-EDF Scheduling Technique.	Based on control pre-emption (cP-EDF), novel preemption approaches that control preempting processes for better "earliest deadline first" schedulability under a specified preemption delay by up to 7.4%.

C. Classification based on Criteria Measure Performance

This section identifies the parameters based on the performance assessment used to analyze scheduling techniques. Existing papers also used a variety of applications of criteria

measures, such as CPU, performance, turn-around time, waiting-time, response-time, fairness, and others. As we have seen from Table III, is the most metric used for assessing the overall performance of the scheduling techniques that were developed [47, 48].

TABLE III. CLASSIFICATION BASED ON ANALYSIS CRITERIA MEASURE PERFORMANCE ASSESSMENT.

Ref. No	Novel Proposed Method with Existing Approaches.	Criteria based Measure Overall Performance for Novelty Proposed						
		CPU Utilize	Throughput	Average Turn-around time	Average Waiting Time	Response Time	Fair	Context Switching
[20]	OIPSA with FCFS, SJF, and RR.	-----	-----	332.2 Ms.	293.84 Ms.	387.8 Ms.	-----	-----
[21]	AN-RR with Traditional - RR.	-----	-----	10.4412 Ms.	10.4413 Ms.	-----	-----	Decreased 50 %
[22]	Improved Dynamic Round Robin IDRR with SRR, DQRRR, SARR, and DABRR	Max. CPU utilize	-----	157 Ms.	97.4 Ms.	-----	-----	Increase 1%
[23]	IODTS-RR with ODTs-RR.	-----	Maximize Throughput	75.08 Ms.	48.06 Ms.	41.73 Ms.	-----	-----
[28]	Two FIS with Traditional MLQS	-----	-----	-----	38 Sec.	8.12 Sec.	Fair with 2 Queue.	-----
[31]	New MLFQ with Traditional MLQS	Max. CPU utilize	Maximize Throughput	107.2 Ms.	72.4 Ms.	28 Ms.	Fairly	-----
[32]	Static & Dynamic Time Quantum		-----	Static Avg. Que3 = 351 Ms.	Static Avg. Que3 = 228 Ms.	-----	-----	Increase No. of CS.

	Implementation MLFQ with Traditional MLFQ.	Max. CPU utilize		Dynamic Avg. Que3 = 252 Ms.	Dynamic Avg. Que3 = 119 Ms.			
[33]	A Simulator Based MLFQ with Two Cases RR, SJFRR	Slightly Max. CPU utilize	-----	Static and Dynamic QT Avg. Case 1=807, 754 Ms. Static and Dynamic QT Avg. Case 2 717, 670 = Ms.	Static and Dynamic QT Avg. Case 1 = 116.2, 107.6 Ms. Static and Dynamic QT Avg. Case 2 98.2, 91.8 = Ms.	-----	RR More Fairly for CPU than SJFRR	Increase Slightly No. of CS.
[39]	Genetic algorithm with SJF and FCFS.	-----	Maximize Throughput	Slightly Minimize ATAT.	Min. 29.8 Ms Maxi. 50.6 Ms GA. Near to SJF.	-----	-----	Reduce No. of CS.

As we are shown in Table IV, it concerns the efficiency of the various scheduling techniques analysis in this paper and the other techniques that apply to the scheduling algorithms discussed in terms of data set and performance. The most criteria based- measures utilize for performance assessment: turnaround

time, waiting time, response time, and context switching are considerably respectively.

TABLE IV. SUMMARIZES DIFFERENT NOVEL SCHEDULING TECHNIQUES PERFORMANCE ASSESSMENT.

Researchers work no.	Novel Proposed Scheduling Techniques.	Performance assessment in this work
[20]	Intelligent Process Scheduling Algorithm, based on an organization.	OIPSA reduced both response-time, wait-time, and turn-around time for preferred processes in the strategy and increases overall system efficiency.
[21]	New Round Robin Scheduling techniques.	Solves the issues of "fixed time quantum," increases the Round Robin's efficiency, minimizes response time, and increases the use of CPUs.
[22]	Round-Robin Scheduling techniques.	Maximum throughput, minimize average turn-around time, waiting-time, and reduce context switching by using the variant quantum time.
[23]	Round Robin Scheduling techniques.	Allocate the CPU equally to each process in order, reducing context-switching, eliminating response time and average waiting time.
[28]	Multilevel Queue Scheduling techniques.	Improved the quality of services and excellently enhances the performance by minimizing response time and allocate CPU fairly to each job.
[31]	New Multilevel Feedback Queue Scheduling techniques for the OS	Eliminate the limitations and solves the problems of response-time.
[32]	Multilevel Feedback Queue Scheduling techniques.	Improve the efficiency of scheduling by minimizing CPU overhead and allocate it's fair to each process.
[33]	Multilevel Scheduling techniques for Feedback Queue.	MLFQ gives students the flexibility to specialize in input process applications by CPU utilize with queuing.
[39]	Genetic algorithm with scheduling strategy FCFS and SJF	The experimental results indicate that the FCFS and SJF with the genetic algorithm increase average cycle average waiting time on a single processor.

Our contribution of this review paper is to provide previous research's works on the scheduling algorithm in the CPU. Also, providing a new proposed approach for them based on criteria measure performance. This review paper can be a robust beginning point for researchers who discuss to optimize or improve scheduling techniques or either develops it in different fields.

V. CONCLUSION AND FUTURE WORK

In this review, the various CPU scheduling strategies are assessed AWT, ATAT, RT, and some other metrics based on the measure of scheduling-criteria. The Advanced technologies are looking towards the advancements in scheduling techniques. This paper focuses on the strategy of implemented and performance assessment in many approaches proposed alternative as reviewed in this paper. It tends to focus on a specific scheduling requirement, for example, round-robin and multilevel input queue scheduling strategies are widely considered to improve performance evaluation of the CPU scheduling algorithm, such as eliminated context switches, and indefinite blocking (starvation) problems. This paper also states

out that every scheduling approach can be based on process scheduling within the OS, as specified by the application. So many other scheduling strategies such SJRTF and priority scheduling technique provide overhead medium CPU, low throughput, medium turnaround and response time.

Future work, it is must find the easiest solution to overcome common limitations of most of the CPU techniques such as starvation problem, number of contexts switching and others. So, this research might be extended by analyzing comparative in many different environments.

REFERENCES

- [1] S. William, *Operating systems : internals and design principles*. 2011.
- [2] L. M. Haji, S. R. M. Zeebaree, K. Jacksi, and D. Q. Zeebaree, "A State of Art Survey for OS Performance Improvement," *Sci. J. Univ. Zakho*, vol. 6, no. 3, pp. 118–123, 2018, doi: 10.25271/sjuz.2018.6.3.516.
- [3] A. S. W. Andrew S. Tanenbaum, *Operating Systems Design and Implementation, Third Edition*, Prentice Hall. 2006.
- [4] M. A. Mohammed, M. Abdulkmajid, B. A. Mustafa, and R. F. Ghani, "Queueing theory study of round robin versus priority dynamic

quantum time round robin scheduling algorithms,” *2015 4th Int. Conf. Softw. Eng. Comput. Syst.*, pp. 189–194, 2015, doi: 10.1109/ICSECS.2015.7333108.

[5] J. Zhou, X. Zhou, Z. Chen, and X. Chen, “Research of real-time control algorithm for traffic lights based on CPU process scheduling,” *2011 IEEE Int. Conf. Anti-Counterfeiting, Secur. Identif.*, pp. 110–114, 2011, doi: 10.1109/ASID.2011.5967428.

[6] H. B. Parekh and S. Chaudhari, “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,” *Int. Conf. Glob. Trends Signal Process. Inf. Comput. Commun.*, pp. 184–187, 2017, doi: 10.1109/ICGTSPICC.2016.7955294.

[7] S. S. Padmashree M G, “S2 R2: An enhanced method to improve the performance of CPU scheduling using Sort-Split Round-Robin Technique for load balancing,” *2016 IEEE*, pp. 854–857.

[8] M. U. Farooq, A. Shakoor, and A. B. Siddique, “An Efficient Dynamic Round Robin algorithm for CPU scheduling,” *Proc. 2017 Int. Conf. Commun. Comput. Digit. Syst. C-CODE*, pp. 244–248, 2017, doi: 10.1109/C-CODE.2017.7918936.

[9] G. C. Buttazzo, M. Bertogna, and G. Yao, “Limited preemptive scheduling for real-time systems. A survey,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 3–15, 2013, doi: 10.1109/TII.2012.2188805.

[10] O. Ahmed and W. Abdullah, “A Review on Recent Steganography Techniques in Cloud Computing,” *Acad. J. Nawroz Univ.*, vol. 6, no. 3, pp. 106–111, 2017, doi: 10.25007/ajnu.v6n3a91.

[11] H. Mora, “Modified Median Round Robin Algorithm (MMRRA),” *2017 IEEE*.

[12] B. Dave, S. Yadav, M. Mathuria, and M. Tech, “Customary Methods for CPU Scheduling : A Review,” vol. 3, no. 8, pp. 344–348, 2017.

[13] A. Silberschatz, P. B. Galvin, and G. Gagne, “Operating System Concepts,” *Ninth Ed. Addison Wesley, Boston*, 2012.

[14] Y. H. Jbara, “A new improved round robin-based scheduling algorithm-a comparative analysis,” *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 1–6, 2019, doi: 10.1109/ICCISSci.2019.8716476.

[15] N. Almansour and N. M. Allah, “A survey of scheduling algorithms in cloud computing,” *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 1–6, 2019, doi: 10.1109/ICCISSci.2019.8716448.

[16] M. Akhtar, B. Hamid, and M. Humayun, “An Optimized Shortest job first Scheduling Algorithm for CPU Scheduling,” *J. Appl. Environ. Biol. Sci.*, vol. 5, no. 12, pp. 42–46, 2015.

[17] F. Ahmad, I. Khan, S. A. Mahmud, G. M. Khan, and F. Z. Yousaf, “Real time Evaluation of Shortest Remaining Processing Time based Schedulers for Traffic Congestion Control using Wireless Sensor Networks,” *2013 Int. Conf. Connect. Veh. Expo*, pp. 381–387, 2013, doi: 10.1109/ICCVE.2013.90.

[18] R. I. Davis and M. Bertogna, “Optimal fixed priority scheduling with deferred pre-emption,” *Proc. - Real-Time Syst. Symp.*, pp. 39–50, 2012, doi: 10.1109/RTSS.2012.57.

[19] J. Chen, W. Huang, Z. Dong, and C. Liu, “Fixed-Priority Scheduling of Mixed Soft and Hard Real-Time Tasks on Multiprocessors,” *2017 IEEE*.

[20] M. A. Shah, M. B. Shahid, S. Zhang, S. Mustafa, and M. Hussain, “Organization Based Intelligent Process Scheduling Algorithm (OIPSA),” *Proc. 21st Int. Conf. Autom. Comput. Univ. Strat. Glas. UK, 11-12 Sept. 2015*, doi: 10.1109/IConAC.2015.7313978.

[21] A. Noon, A. Kalakech, and S. Kadry, “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average,” *J. Comput. Sci. Issues*, vol. 8, no. 3, pp. 224–229, 2011.

[22] A. Alsheikhy, R. Ammar, and R. Elfouly, “An improved dynamic Round Robin scheduling algorithm based on a variant quantum time,” *2015 11th Int. Comput. Eng. Conf. Today Inf. Soc. ICENCO*, pp. 98–104, 2015, doi: 10.1109/ICENCO.2015.7416332.

[23] S. Ullah, “Improved Optimum Dynamic Time Slicing Round Robin Algorithm,” *2017 3rd Int. Conf. Electr. Inf. Commun. Technol.*, no. December, pp. 7–9, 2017.

[24] L. Luo, W. Wu, D. Di, F. Zhang, Y. Yan, and Y. Mao, “A resource scheduling algorithm of cloud computing based on energy efficient optimization methods,” in *2015 IEEE 12th International Conference on e-Business Engineering 23-25 Oct. 2015*, no. June, doi: 10.1109/ICEBE.2015.68.

[25] P. Sangwan, M. Sharma, and A. Kumar, “Improved Round Robin Scheduling in Cloud Computing,” *Adv. Comput. Sci. Technol.*, vol. 10, no. 4, pp. 639–644, 2017.

[26] P. Patil and C. Borse, “Fair resource allocation to MIMO wireless system using Opportunistic Round Robin scheduling algorithm,” *2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc.*, pp. 1–3, doi: 10.1109/PERVATIVE.2015.7087050.

[27] V. Chahar and S. Raheja, “Fuzzy based Multilevel Queue Scheduling Algorithm,” *Proc. 2013 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI*, pp. 115–120, 2013, doi: 10.1109/ICACCI.2013.6637156.

[28] A. Rezaee, A. M. Rahmani, S. Adabi, and S. Adabi, “A fuzzy algorithm for adaptive multilevel queue management with QoS feedback,” *Proc. 2011 Int. Conf. High Perform. Comput. Simulation, HPCS 2011*, pp. 121–127, 2011, doi: 10.1109/HPCSim.2011.5999815.

[29] S. Raheja, R. Dadhich, and S. Rajpal, “2-Layered Architecture of Vague Logic Based Multilevel Queue Scheduler,” *Appl. Comput. Intell. Soft Comput.*, vol. 2014, 2014.

[30] S. M. F, “Broadcast Network (IRBN) Over Satellite,” *Int. Conf. Intell. Comput. Control Syst. ICICCS 2017- IEEE*, pp. 145–148, 2017.

[31] S. Kadry and A. Bagdasaryan, “New MLFQ Scheduling Algorithm for Operating Systems Using Dynamic quantum,” *Stat. Optim. Inf. Comput.*, vol. 3, no. 2, pp. 189–196, 2015, doi: 10.19139/soic.v3i2.58.

[32] M. Thombare, R. Sukhwani, P. Shah, S. Chaudhari, and P. Raundale, “Efficient implementation of Multilevel Feedback Queue Scheduling,” *Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016*, pp. 1950–1954, 2016, doi: 10.1109/WiSPNET.2016.7566483.

[33] S. K. Dwivedi and R. Gupta, “A simulator based performance analysis of multilevel feedback queue scheduling,” *Proc. - 5th IEEE Int. Conf. Comput. Commun. Technol. ICCCT 2014*, pp. 341–346, 2015, doi: 10.1109/ICCT.2014.7001516.

[34] K. E. Hoganson and D. Ph, “Multi-Core Real-Time Scheduling in Multilevel Feedback Queue with Starvation Mitigation (MLFQ-RT),” *ACM SE '18, March 29–31, 2018, Richmond, KY, USA*.

[35] S. Killen, E. Giese, and H. Huynh, “Marble MLFQ : An Educational Visualization Tool for the Multilevel Feedback Queue Algorithm,” *2017 IEEE*, pp. 663–669.

[36] D. S. J. Shweta Jain, “Analysis of Multi Level Feedback Queue Scheduling Using Markov Chain Model with Data Model Approach,” *Int. J. Adv. Netw. Appl.*, vol. 07, no. 06, pp. 2915–2924, 2016.

[37] A. Patil and R. V. Biradar, “Scheduling techniques for TinyOS: A review,” *2016 Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS*, pp. 188–193, 2016, doi: 10.1109/CSITSS.2016.7779420.

[38] A. O. Chhabra, “Analytical Study of Job Scheduling using Variants of Ant Colony Optimization Technique in Grid,” *2nd IEEE Int. Conf. Eng. Technol.*, no. March, pp. 16–21, 2016.

[39] T. A. Maktum, R. A. Dhumal, and L. Ragha, “A genetic approach for processor scheduling,” *Int. Conf. Recent Adv. Innov. Eng. ICRAIE*, pp. 9–12, 2014, doi: 10.1109/ICRAIE.2014.6909108.

[40] S. Jawad, “Design and evaluation of a neurofuzzy CPU scheduling algorithm,” *Proc. 11th IEEE Int. Conf. Networking, Sens. Control. ICNSC*, pp. 445–450, 2014, doi: 10.1109/ICNSC.2014.6819667.

[41] T. Helmy, S. Al-Azani, and O. Bin-Obaidullah, “A machine learning-based approach to estimate the CPU-burst time for processes in the computational grids,” *3rd Int. Conf. Artif. Intell. Model. Simul.*, 2015, doi: 10.1109/AIMS.2015.11.

[42] K. S. Umadevi, “Multilevel Queue Scheduling in Software Defined Networks,” *Int. Conf. Innov. Power Adv. Comput. Technol.*.

[43] S. Kumarsaroj, A. K. Sharma, and S. K. Chauhan, “A novel CPU scheduling with variable time quantum based on mean difference of burst time,” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2016*, no. April, pp. 1342–1347, 2017, doi: 10.1109/CCA.2016.7813986.

[44] J. Lee and K. G. Shin, “Preempt a job or not in EDF scheduling of uniprocessor systems,” *IEEE Trans. Comput.*, vol. 63, no. 5, pp. 1197–1206, 2014, doi: 10.1109/TC.2012.279.

[45] Y. H. Jbara, "A new visual tool to improve the effectiveness of teaching and learning CPU scheduling algorithms," *2017 IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. AEECT 2017*, vol. 2018-Janua, pp. 1–6, 2017, doi: 10.1109/AEECT.2017.8257759.

[46] O. Ahmed and A. Brifcani, "Gene Expression Classification Based on Deep Learning," *4th Sci. Int. Conf. Najaf, SICN 2019*, pp. 145–149, 2019, doi: 10.1109/SICN47020.2019.9019357.

[47] A. R. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," *Futur. Gener. Comput. Syst.*, vol. 91, pp. 407–415, 2018, doi: 10.1016/j.future.2018.09.014.

[48] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and S. R. M. Zeebaree, "Combination of k-means clustering with genetic algorithm: A review," *Int. J. Appl. Eng. Res.*, vol. 12, no. 24, pp. 14238–14245, 2017.