

The Effect of Context Switching on Multilevel Feedback Queue (MLFQ) Scheduling Algorithms Performance and CPU Utilization

AL REFAI MOHAMMED N.
Software Engineering Department,
Zarqa University,
Zarqa,
JORDAN

ORCID: <https://orcid.org/0000-0002-3464-8900>
Scopus Author ID: 57213988453

Abstract: - One of the most important jobs of an operating system is to schedule processes. The most important thing is to make sure that a lot of processes in the RAM's ready queue are using the CPU at once. You should also monitor CPU usage, response time, wait time, and turnaround time. This paper looks at how changing contexts affect the scheduling process in the Multi-Level Feedback Queue (MLFQ) system. The goal is to determine how often context switches occur. With that information, we can then figure out response time, CPU usage, and data transfer. The study also looks at how to fix negative effects, such as making things more flexible and reducing unnecessary context switching. The main goal of this paper is to help people who design and implement operating systems use resources more efficiently when many tasks are running at once. They will find the results very helpful. The main goal of this paper is to improve the system and make it easier for people to use.

Key-Words: - operating system, scheduling algorithms quantum, multilevel feedback queue, context switching, optimization, and CPU utilization.

Received: April 28, 2026. Revised: June 15, 2026. Accepted: July 19, 2026. Published: October 20, 2026.

1 Introduction

Context switching is very important in modern operating systems because it lets you do more than one thing at a time and use resources better [1]. It is the act of saving the current state of a running process so the operating system can switch to a different process and keep it running. But changing contexts too often can slow down the system a lot, especially when it comes to scheduling algorithms [2].

The Multi-Level Feedback Queue (MLFQ) algorithm is a common way to set up tasks. This algorithm uses a number of queues, each of which is more or less important than the others. Processes can move from one queue to another based on their behavior and needs [3]. The goal of this flexible method is to make sure that processes with different characteristics are treated fairly and quickly [4]. The MLFQ algorithm might only work well and be useful when it changes contexts [2], [3].

This year, researchers looked into MLFQ to see if it could help them organize their work better. The authors of [5] proposed an MLFQ algorithm that uses a fixed time quantum for the first queue. They proposed a scheduling algorithm that uses a dynamic time quantum for the second queue. The research paper in [6] asserts that the modified MLFQ method surpasses current scheduling algorithms. It also lets short, important tasks wait less time. The study in [7] suggested a fuzzy-based task offloading technique (FBOT) to find the best way to assign tasks that are heavy on data and computation to the right fog-cloud system. This paper examines the impact of context switching on the performance of MLFQ scheduling systems. They examine various pertinent metrics, such as CPU utilization, throughput, and response time, to assess the impact of context switches on the algorithm's performance. This study will use both synthetic workloads and real applications to ensure the results are useful in a variety of situations.

We will brainstorm and perform experiments with the regulation of the frequency of context switches; if necessary, we will use complex performance counters to reach the goals mentioned above [2]. Experiments will be performed on a testbed with the usual operating system and various workloads [1]. The change in the context-switching paradigm caused by the MLFQ algorithm makes it difficult to detect such settings and tendencies based on observations and comparisons of performance metrics.

This research will also analyze ways to minimize the negative consequences of context switches on MLFQ execution. Such research can consider minimizing context switching [2], [8], increasing the ability of an OS to adapt to changing workloads [9], or searching for new techniques to manage context switching [8].

The main goal of this research is to determine the benefits and drawbacks of MLFQ scheduling and context switching. Knowing about the negative effects of context switching and its impact on performance, developers can more efficiently approach meeting scheduling in designing and implementing operating systems. This knowledge may prove beneficial, as it may enable the efficient use of resources and, consequently, increase the performance and usability of the entire system and support further development.

2 Metrics for Simulations

The simulation of the Multilevel Feedback Queue (MLFQ) scheduling algorithm was carried out to analyze the effects of context switches on various performance measures such as quantum, average response time, average waiting time, and average turnaround time [10]. MLFQ utilizes the Round Robin (RR) scheduling algorithm in each level and the FIFO scheduling algorithm for promotion of processes in case of an increase in levels [11].

Various performance measures were analyzed for their effects on the MLFQ scheduling algorithm in case of change in context:

- a. Quantum: Quantum is the quantum time slice assigned to each process in each level of the MLFQ [12]. The effects of changing the quantum time are observed with varying performance measure values.
- b. Average Response Time: The average response time refers to the time that the process requires to reach the system and start execution. It represents the speed at which processes receive CPU time, making it an ideal metric for measuring system performance.
- c. Average Waiting Time: Average waiting time is the time that a process spends in the ready queue before obtaining CPU time [3]. In this case, the efficiency of the scheduling algorithm regarding handling process queues can be established.
- d. Average Turnaround Time: Average turnaround time refers to the time required by a process to be completed. It entails both CPU processing time and the time required to obtain it [13], [14]. Therefore, the performance of the system can be assessed using the metric.

By varying the quantum value in the simulation, the frequency was varied, hence enabling the observation of changes in all the above metrics. The purpose of this assignment was to design the simulation program to record such metrics [10].

It would be possible to observe the impact of altering the context on the metrics in the MLFQ simulation (RR, RR, FIFO). We will be able to understand from the results of the simulation how MLFQ performs more slowly and reacts less to the changes in the context.

3 Setting Up the Scenario

The experiment for determining the effect of quantum values on the number of context switches was carried out using the MLFQ algorithm for scheduling 100, 1000, and 10,000 processes. In this case, changes were made to quantum values in order to analyze their impact on the ability to switch contexts. The objective of this research was to investigate how quantum values affect context switching under various workloads through simulations of varying process numbers. In particular, 100, 1000, and 10,000 processes were chosen in order to demonstrate the effect of quantum values on workloads with varying intensities [1].

We applied the MLFQ algorithm to calculate the frequency of context switches during the simulation for all combinations of the quantum value and process counts. This research was designed to analyze patterns or trends in the relationship between

quantum and the frequency of context switches by modifying the quantum value [2].

Other aspects of the simulation were also considered to ensure accuracy, including the sequence of process arrivals, CPU burst durations, and priorities. These aspects were chosen with precision to resemble the actual environment in which they occur, in an effort to understand the workings of the MLFQ algorithm under certain quantum values and workloads [8].

To find out how the quantum value influences the number of context switches during the operation of the MLFQ algorithm, the researchers modified both the quantum value and the number of processes in the simulation. Such findings can be quite useful in planning and improving the system [3]. Figure 1 illustrates the effect of the process undergoing the first simulation configuration with a quantum of 3 for the first RR queue and 6 for the second round-robin (RR) queue.

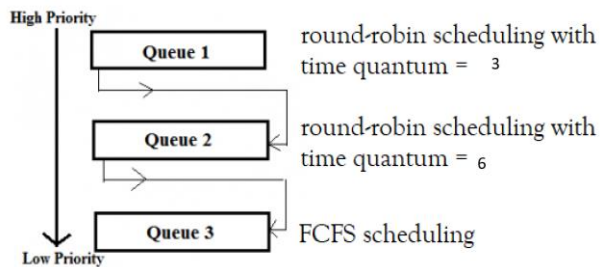


Fig. 1: First Simulation Setup

Source: created by the authors

In our second simulation setup, Figure 2 shows how the process works with a quantum of 13 for the first RR queue and 3 for the second round-robin (RR) queue.

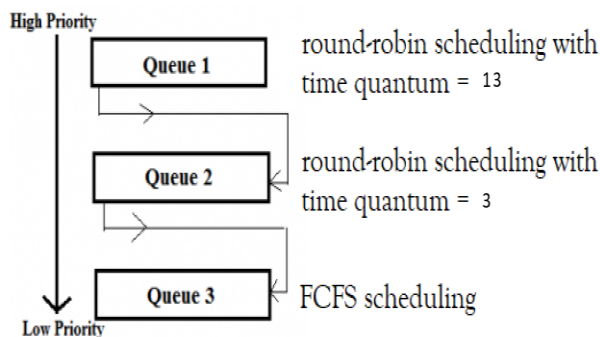


Fig. 2: Second Simulation Setup

Source: created by the authors

4 Results of the Simulation

When 100 processes were run through a simulation, this is what happened. The first round-robin (RR) queue had a quantum of 3, and the second round-robin (RR) queue had a quantum of 6 as shown in Figure 3.

```
Average response time: 8.54
Average waiting time: 11.61
Average turnaround time: 13.1
Number of Context Switching: 171
CPU Utilization :68.10345%
```

Fig. 3: Simulation Results for 100 Processes Case 1

Source: created by the authors

Figure 4 shows what happened when 100 processes were run on the same data set. The first round-robin (RR) queue had a quantum of 13, while the second queue had a quantum of 3.

```
Average response time: 9.98
Average waiting time: 11.72
Average turnaround time: 11.36
Number of Context Switching: 121
CPU Utilization :68.10345%
```

Fig. 4: Simulation Results for 100 processes Case 2

Source: created by the authors

When using 100 processes, Figure 5 shows how the performance metrics differ between the two algorithms. As quantum time increases, response time and turnaround time decrease, but the number of context switches increases.

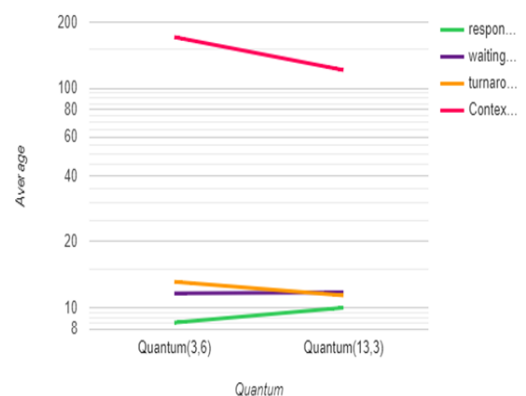


Fig. 5: Metrics Comparations 100 processes

Source: created by the authors

The simulation result of 1000 processes with a quantum of 3 for the first round-robin (RR) queue and 6 for the second queue is presented in Figure 6.

```
Average response time: 8.395
Average waiting time: 12.145
Average turnaround time: 13.863
Number of Context Switching: 1660
CPU Utilization :67.942764%
```

Fig. 6: Simulation Results for 1000 processes case 1
Source: created by the authors

Figure 7 shows the results of a simulation with 1000 processes, where the first round-robin (RR) queue had 13 items and the second queue had 3 items.

```
Average response time: 9.99
Average waiting time: 12.435
Average turnaround time: 12.648
Number of Context Switching: 1200
CPU Utilization :67.942764%
```

Fig. 7: Simulation Results for 1000 processes case 2
Source: created by the authors

Figure 8 shows how the two algorithms' performance metrics are different when 1000 processes are used. As quantum time increases, the number of context switches decreases.

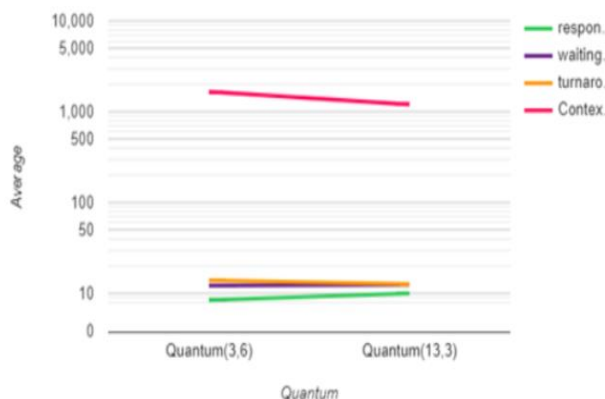


Fig. 8: Metrics Comparisons 1000 processes
Source: created by the authors

Figure 9 shows the results of a simulation of 10,000 processes, with 3 for the first round-robin (RR) queue and 6 for the second queue.

```
Average response time: 85.471
Average waiting time: 125.991
Average turnaround time: 141.742
Number of Context Switching: 16618
CPU Utilization :66.67644%
```

Fig. 9: Simulation Results for 10000 processes case 1
Source: created by the authors

The simulation result in Figure 10 shows the results of a simulation with 10,000 processes, a quantum of 13 for the first round-robin (RR) queue, and a quantum of 3 for the second queue on the same data set.

```
Average response time: 102.396
Average waiting time: 128.335
Average turnaround time: 131.107
Number of Context Switching: 11676
CPU Utilization :66.67644%
```

Fig. 10: Results of the simulation for 10,000 processes, Case 2
Source: created by the authors

When using 10000 processes, Figure 11 shows how the performance metrics for the two algorithms are different. The main point is that as quantum time decreases, the number of context switches increases. The next two sections contain analysis and discussion.

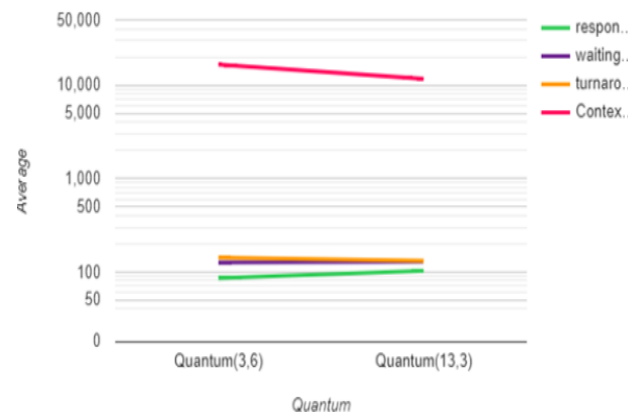


Fig. 11: Metrics Comparisons 10,000
Source: created by the authors

5 Result Analyze

The discussion focuses on how changing the quantum value in the MLFQ algorithm affects context switching and the importance of such changes.

Each process is allocated a quantum value in an assumed queue before a context switch takes place. By augmenting the quantum value, the scheduler permits each process to execute longer before preempting it and transforming to another process. This leads to a decline in the occurrence of context switching.

Diminishing the number of context switches can yield several benefits. Firstly, context switching is a time-consuming operation that presents overhead and can impact the overall performance of the system [1]. By minimizing context switches, the system can devote more time to the actual execution of processes, resulting in enhanced efficiency.

Furthermore, frequent context switches may disrupt the execution flow of processes, resulting in increased response time and waiting time [2]. When the quantum value is increased, processes have a longer uninterrupted execution time, reducing the requirement for frequent context switches and improving response time.

However, one must take into account the possible sacrifices that come along with this choice. Increasing the quantum value could minimize context switches, but it could also increase turnaround time for some processes. Processes with lower burst times will experience longer wait times before they can execute again, which can result in waiting times.

The importance of adjusting the context switching process comes from finding a balance between minimizing context switches and other performance metrics. Through adjustment of the quantum value, adjustments in scheduling algorithms can be made [14]. In time-critical processes, a smaller quantum value may work better in order to minimize response time through greater context switching.

For processes involving heavy CPU use, a larger quantum value should be implemented to minimize the overhead cost associated with reduced context switching.

In summary, the impact of modifying quantum value on context switching highlights the importance of taking careful note of performance metrics and the needs of the particular system. Ensuring the balance

between context switching and other performance metrics is vital to the effectiveness of scheduling algorithms.

6 Discussions

This discussion will be focused on how changing quantum values in the MLFQ algorithm affects context switching and its significance.

Quantum values determine the time allotted to a process within a certain queue until a context switch is performed [15]. The quantum value in the algorithm can be increased, ensuring that every process runs for a longer time before a context switch to a different process. This results in fewer context switches.

There are various benefits of minimizing context switching. First, context switching is time-consuming and thus reduces efficiency [16]. As a result, minimizing context switching ensures that processes have sufficient time to execute rather than performing unnecessary switching procedures.

Furthermore, frequent switching may affect the progress of processes as they wait before they continue executing, which means that they will increase their waiting and response time [17]. Increasing quantum values will allow processes to run without frequent interruptions from context switches.

But there are considerations to take into account when considering this change. While increasing the quantum would reduce context switches, turnaround time for some processes would also increase. This is especially the case when dealing with processes that have short burst time since they will have to wait longer to execute their CPU usage.

Changing the context-switching quantum in scheduling algorithms is considered vital to find the best trade-off between reducing context switches and maintaining optimal performance in other performance measures. By fine-tuning the quantum size, the designer can make changes based on their own needs.

In real-time systems where the objective is to reduce response time, a low quantum size should be preferred to allow frequent context switching and hence quick process execution. On the other hand, in CPU-bound applications where overhead reduction is essential, a high quantum should be used in order to avoid context switches.

Generally, changing the quantum value in context switching calls for careful consideration of several performance measures, among others.

6.1 Quantum Value's Impact on Context Switching

In analyzing the relationship between quantum value and context switching in the Multi-Level Feedback Queue (MLFQ), the role of modifying the quantum value is considered [18], [19]. Quantum value is the gate that determines the amount of time a process takes before a context switch takes place [20].

As such, modifying the quantum value increases the execution time for processes prior to context switching, thus reducing the rate at which context switching occurs [18]. There are many benefits associated with fewer context switches; the most important of all is releasing critical system resources that context switching requires [19].

Context switching interrupts the process execution flow, hence causing delays in both response and waiting times [21]. However, increasing the quantum value ensures that the processes execute without context switching, which improves the overall response times of processes in the system [11].

Nevertheless, care should be taken to maintain a delicate balance, taking into consideration the possible trade-offs that may result from a high quantum value, and this may increase the turnaround time of some processes [22]. Striking a perfect balance is essential in this regard.

The following Table 1 and Figure 12 illustrate the effect of quantum values (RR1, RR2, and FIFO) on context switching, along with data regarding the number of context switches and the number of processes.

Table 1. Context Switching for All Cases in Simulation

	100	1000	10000
(3, 6)	171	1660	16618
(13, 3)	121	1200	11676

Source: created by the authors

6.2 Quantum Value Dynamics and Performance

6.2.1 Optimization

However, tampering with quantum values has implications beyond context switching, influencing

system performance in several different ways. An increase in the quantum value would lead to a reduction in the number of context switches, allowing for better use of CPU resources and thus improving efficiency [22].

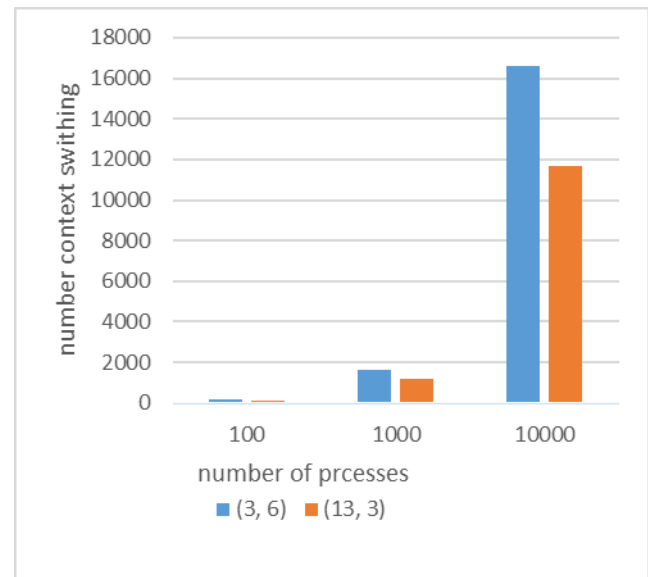


Fig. 12: Relationship between number of processes and the number context switching

Source: created by the authors

This holds specific significance in environments characterized by high throughput, where the decrease in the overhead involved in the process of context switching is important in ensuring optimal performance of the system [23]. As an example, in the case of a server that is bombarded with difficult requests for processing, use of a larger quantum enables more processing of the tasks prior to any context switching, thus reducing the overheads associated with context switching [14].

Nonetheless, the effect of quantum size goes beyond context switching, as it also influences the optimization of response times [24]. Smaller quantum sizes lead to quicker decisions and increased response capabilities of the system, just as a diligent waiter quickly responds to customers' requests in a restaurant [25].

However, where there is a greater quantum, the process of giving preference to the throughput process will be made easier through limiting the number of times context switches occur, especially in highly intensive computational environments such as data centers.

The advantage of MLFQ's adaptability is that the quantum value can be easily adjusted by the designer according to the nature of the workload in question, making it possible for the system to perform optimally irrespective of the type of operations taking place in it.

6.2.2 Practical Considerations and Recommendations

After analyzing the influence that quantum value has on the process of switching from one context to another and on the whole system operation, it is high time to provide some recommendations that may prove useful to specialists working with systems optimization.

6.2.3 Refining Quantum Values:

Matching Quantum Value with Workload Characteristics: It is essential to match the quantum value with the workload characteristics. If the CPU is a limiting factor for a process, an increased quantum value would help reduce the extra burden of context switching, thus increasing overall throughput.

Table 2. Optimization Recommendations for MLFQ Scheduling

Recommendation	Details	Potential Benefits
Fine-Tuning Quantum Values	Match quantum value to workload characteristics	Improved system performance
	Periodic evaluation of quantum values	Adaptation to changing workloads
Balancing Resource Allocation	Optimize resource allocation	Efficient use of system resources
	Implement starvation mitigation	Fair process execution
Visualization and Monitoring	Monitor performance metrics	Data-driven decision-making
	Historical analysis and trend prediction	Informed quantum value adjustments

Source: created by the authors

On the other hand, in situations that require quick response from the system, a lower quantum value should be used.

Evaluation of Quantum Values: Quantum values are not always static. In fact, they need to be changed

with respect to changes in workloads. Thus, it is very important to evaluate the quantum values.

6.2.4 Balancing Resource Allocation

Efficient Use of Resources in Scheduling: Ensuring proper trade-off between context switching and resource distribution is critical. Although fewer context switches contribute to efficient operations, this must be done without creating any possible problem of bottleneck or delay. Proper consideration is necessary to determine the right quantum so that effective resource allocation can be achieved.

Addressing Process Starvation: It will be particularly important to address process starvation, especially if large quantum numbers are being used. The processes with short burst times should not experience any delay.

6.2.5 Visualization and Monitoring:

Constant monitoring of performance metrics, such as the number of context switches, response time, and waiting time, is strongly suggested. Visualization can help monitor changes after adjusting the quantum value. A data-driven approach will allow for better decision-making. Using historical information and tendency analysis can help in assessing how modifications to the quantum value can affect system performance in the future.

Following the suggestions in Table 2 and allowing the use of a data-driven approach for changing the quantum value in the MLFQ algorithm will help achieve the right balance between the efficiency of context switching and optimal system performance.

7 Conclusion

MLFQ algorithm requires quantum values to function effectively. The quantum value influences many performance indicators like system throughput, turnaround time, and context switching. Context switching plays a significant role in performance of any system since this refers to moving of CPU from one process to another and maintaining the process state. The findings of the research indicate that the quantum value enables one to manipulate the above figures. For example, choosing a smaller quantum value can make the average wait time shorter and the system more efficient, but it will also cost more to context switches [19]. On the other hand, picking a

longer quantum value makes it cheaper to switch contexts, but it also makes the average wait time longer and the system's Being fair. So, it's very important to find a balance between lowering context switching and reaching our performance goals. Changing the quantum value is a very important step in getting the MLFQ algorithm to work better for us and the whole system to work better.

Acknowledgment:

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the research.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] E. El-Sharawy, "A Review on the CPU Scheduling Algorithms: Comparative Study," *International Journal of Network Security*, vol. 21, no. 13, 2021, pp. 19-26, DOI: 10.22937/IJCSNS.2021.21.1.4.
- [2] S. & B. L. & A. M. Zouaoui, "CPU scheduling algorithms: Case & comparative study," in *17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Tunisia, 2016.
- [3] K. Hoganson, "Reducing MLFQ scheduling starvation with feedback and exponential averaging," *Journal of Computing Sciences in Colleges (JCSC)*, vol. Vol. 25, no. Issue 2, pp. 196–202.
- [4] Yadav, Pratibha and D. P. Vidyarthi, "An efficient fuzzy-based task offloading in edge-fog-cloud architecture," *Concurrency and Computation: Practice and Experience*, vol. Vol. 35, Issue 26, p. DOI: 10.1002/cpe.7843, 2023.
- [5] A. . R. Maria and B. Rajkumar , "A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments," *Concurrency and Computation: Practice and Experience*, vol. 9, no. 1, p. DOI: 10.1002/cpe.4041.
- [6] Kashif, M. Nawaz, R. Rab and Riri, "Multilevel feedback queue: Efficient scheduling and implementation by using dynamic quantum.," in *The 8th International Conference And Workshop On Basic And Applied Science (ICOWOBAS)*, Surabaya, Indonesia, 2023. DOI:10.1063/5.0106240.
- [7] D. M. Dhamdhare, *Operating Systems: A Concept-Based Approach*, 3rd ed., New Delh: McGraw-Hill Education IK Pub, 2012. ISBN-10: 1259005585, ISBN-13: 978-1259005589 · Edition. 3rd · Publisher. McGraw Hill.
- [8] S. Bansal and H. P. , "Analysis of Modified Multilevel Feedback Queue Scheduling Algorithm," *International Journal of Computer Applications*, vol. 73, no. 16, 2013.
- [9] M. Kashif, R. Nawaz, R. Rulaningtyas and S. Iqbal, "Multilevel feedback queue: Efficient scheduling and implementation by using dynamic quantum," in *AIP Conference Proceedings (Vol. 2554, No. 1)*. AIP Publishing., Surabaya, Indonesia, doi: 10.1063/5.0106240, 2023.
- [10] N. Fu, L. Shan, C. Du and Z. Liu, "Modelling and Verification of ARINC 653 Hierarchical Preemptive Scheduling," *The International Arab Journal of Information Technology (IAJIT)*, vol. 17, no. 1, 2020, pp. 99-106, doi: 10.34028/iajit/17/1/12.
- [11] M.-K. Y. e. al, "SecureCore: A Multicore-Based Intrusion Detection Architecture for Real-Time Embedded Systems," Berlin, Germany, 2014, DOI: 10.1109/RTAS.2013.6531076.
- [12] A. M. T. K. N. Saha, "Hybrid CPU Scheduling Algorithm for Operating System to Improve User Experience," in *Kaiser, M.S., Xie, J., Rathore, V.S. (eds) ICT: Smart Systems and Technologies. ICTCS 2023*, Singapore, 2024,

- DOI: doi.org/10.1007/978-981-99-9489-2_38.
- [13] Silberschatz, G. P. A. and G. Gagne, Operating System Concepts (10th ed.), 10 th ed., Wiley, April 2018. ISBN: 978-1-119-80036-1.
- [14] W. Stallings, Operating Systems: Internals and Design Principles, 9th ed., 9 ed., Boston, MA: Pearson, 2018. ISBN-13: 978-0134670959.
- [15] Dr.S.Selvi and O. G. Elamathy , "Enhancing The Performance Of Task Execution In Multicore System: The Impact Of Multi-Level Feedback Queue On Round Robin Adaptive Priority Scheduling," *International Journal of Multidisciplinary Educational Research*, vol. 12, no. 12, pp. 20-29, 2023, [Online]. [https://s3-ap-southeast-1.amazonaws.com/ijmer/pdf/volume12/volume12-issue12\(1\)/3.pdf](https://s3-ap-southeast-1.amazonaws.com/ijmer/pdf/volume12/volume12-issue12(1)/3.pdf) (Accessed Date: January 9, 2026).
- [16] . M. R. EffatParvar , . K. Faez and M. Zarei, "An Intelligent MLFQ Scheduling Algorithm (IMLFQ) with fault tolerant mechanism. Proceedings," in *Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China, 2006. <https://doi.org/10.1109/ISDA.2006.10>.
- [17] S. Tang, S. Boroojerdi, B. Stephen, B. Brough, T. Parker, Jingpeng, J. Walton, D. Sempala and Mulder, "Impact of Quantum Values on Multilevel Feedback Queue for CPU Scheduling," in *2020 Intermountain Engineering, Technology and Computing (IETC)*, Stanford University, 2020. doi: 10.1109/IETC47856.2020.9249188.
- [18] Majedkan, A. Naji, A. Haji and Lailan, "CPU Scheduling Techniques: A Review on Novel Approaches Strategy and Performance Assessment," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 48-55, DOI: 10.38094/jastt1215, 2020.
- [19] Stan, G. Roxana, L. Băjenaru, C. Negru and P. Florin, "Evaluation of Task Scheduling Algorithms in Heterogeneous Computing Environments," *Recent Advances in Big Data and Cloud Computing (MDPI)*, vol. 21, no. 17, pp. 5906 , 2021, DOI: 10.3390/s21175906.
- [20] N. &. K. N. Walia, "Performance Analysis of the Task Scheduling Algorithms in the Cloud Computing Environments," in *2nd International Conference on Intelligent Engineering and Management (ICIEM)*, , 2021. London, United Kingdom, 2021, pp. 108-113, doi: 10.1109/ICIEM51511.2021.9445320.
- [21] A. S. Tanenbaum and H. Bos, Modern Operating Systems (4th ed.), 4 ed., India: Pearson, 2015. ISBN-13: 978-0133591620.
- [22] M. A. Al-Qunaieer, "A comparative study of CPU scheduling algorithms," *International Journal of Graphics & Image Processing*, Vol. 2, Issue 4, November 2012, <https://arxiv.org/pdf/1307.4165>.
- [23] S. K. Dwivedi and R. Gupta , "A simulator based performance analysis of multilevel feedback queue scheduling," Allahabad, India, 2014. pp. 341-346 <https://doi.org/10.1109/ICCCT.2014.7001516>.
- [24] W. Hantom, A. Aldweesh, R. Alzaher and A. ur-Rahman, "A Survey on Scheduling Algorithms in Real-Time Systems," *IJCSNS International Journal of Computer Science and Network Security*,, vol. VOL22, no. No.4,, pp. 686-690. DOI: 10.22937/IJCSNS.2022.22.4.80.
- [25] B. Abirami and V. Vasudevan, "Modified Multilevel Feedback Queue Scheduling Algorithm with Starvation Mitigation for Reconfigurable Computing systems.," *Soft Computing, Springer*, April 2023, DOI: doi.org/10.21203/rs.3.rs-2820144/v1.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself:

This research is funded by the Deanship of Research and Graduate Studies at Zarqa University, Jordan.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US