Discrete Event Simulation of a Commercial Computer with RR, MLFQ, SPN, and HRRN Scheduling Algorithms

NhatMinh Nguyen   
*CS 475W*  
*Whitworth University*Spokane, United States  
nhatminhnguyen22@my.whitworth.eduTsering Ngoche   
*CS 475W*  
*Whitworth University*Spokane, United States  
tngoche23@my.whitworth.edu

*Abstract*—This research investigates the optimization of process scheduling in commercial computer systems through discrete event simulation. Focusing on prioritizing response time and wait time, critical for user satisfaction and system efficiency in environments with high demand for I/O operations, we explored the efficacy of process scheduling algorithms including Round Robin, Multilevel Feedback Queue, Shortest Process Next, and Highest Response Ratio Next. By varying parameters such as time quantum for Round Robin scheduling, we assessed system performance under diverse workloads. Our findings underscore the importance of selecting appropriate algorithms and parameter configurations to achieve optimal performance, shedding light on the complex dynamics of process scheduling in commercial computing environments and contributing to enhanced system efficiency and user experience.

Keywords—quantum, discrete event simulation, starvation, switch time

# Introduction

Back in the old days, the scheduling algorithm was simple, executing the next job on the tape. The more advanced the computers, the more requirements there will be for computers. The requirements like “fast computing, multitasking (execute more than one process at a time) and multiplexing (transmit multiple flows simultaneously) raise the need of scheduling algorithms”[1]. When there are multiple processes, scheduling is a fundamental operating system function that decides which process will run next to assure fairness and avoid the starvation of the processes. There are many scheduling algorithms out there but all of them have to be based on First Come First Serve (FCFS), Round Robins (RR), Shortest Process Next (SPN), and Multilevel Feedback Queues (MFQ), Highest Response Ratio Nest (HRRN). “Scheduling is an ongoing reactive process where the presence of real-time information continually forces reconsideration and revision of pre-established schedules” [2]. This is why the scheduling algorithm is always in consideration for research to find the most optimal for the current system. The research purpose of this paper is to find the most optimal time quantum for Round Robins, the best arrangement of Multilevel Feedback Queues, and compare the results of Round Robins, Multilevel Feedback Queue, Shortest Process Next, and Highest Response Ratio Next to find the most effective algorithms for commercial CPU.

# Scheduling Objectives and Criteria

## Throughput

The number of processes that are completed per time unit [3]. For long processes, it can be one per hour and for short processes would be how many each second. Maximizing the throughput allows as many processes to be completed as possible.

## CPU Usage

The CPU utilization can range from 0 to 100 percent, in a real system, it should range from 40 to 90 percent [3]. Make sure that the percentage of CPU is always above 40% to keep the CPU effective.

## Turnaround Time

Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O [3]. This time is used to see how long the process takes from the time of submission to the time it is complete, or in other words, how long the process takes to execute. Minimizing the turnaround time will make sure the process is accomplished in a short amount of time, resulting in an effective algorithm.

## Waiting Time

This is the time when processes are spent waiting in the ready queue. The longer the waiting time is, the less effective the schedule is. At this time the CPU just sits idle. All this waiting time is wasted; no useful work is accomplished [3].

## Response Time

The time from the submission of a process until the first response from the processor [3]. The lower this amount the faster the processes get executed.

## Time of Completion

This is the time according to the clock at the time the process is finished. The lower this time is the better.

For the goal to find the most optimized quantum for RR and most effective MFQ and comparing with SPN and HRRN, the algorithms would need to maximize CPU utilization and throughput and minimize turnaround time, waiting time, and response time.

# Advantages and disadvantages of rr, mfq, spn, and hrrn

## Round Robins

* Advantage: Simplest, fair for all of the processes, and most widely used in scheduling algorithms[5].
* Disadvantage: If the quantum is extremely large, it will become similar to FCFS and cause the response time to be too large. On the other hand, if the quantum is too small, it will lower the CPU efficiency and cause too many context switches.

## Multilevel Feedback Queues

* Advantage: More flexibility, prevent starvation by changing the priority of a process. By segregating processes into different queues based on their priority levels, MLFQ effectively reduces the occurrence of starvation for long-running processes [9].
* Disadvantage: One potential drawback lies in the complexity associated with determining the optimal number of queues and the appropriate quantum for each queue. The responsiveness and efficiency of MLFQ are directly influenced by these parameters, necessitating careful calibration to achieve optimal performance [9].

## Shortest Process Next

* Advantage: The overall average of waiting time is shortest[5]. The shortest process would be able to dispatch first, allowing more processes to be finished. By prioritizing tasks based on their shortest processing time, SPN enhances resource utilization, allowing for more sophisticated processes to be completed within short time frames[10].
* Disadvantage: The longer processes might have starvation because the shortest processes keep being executed. Another problem is that this algorithm requires the knowledge of how long a process will run and this information is usually unpredictable in real life.

## Highest Response Ratio Next

* Advantage: Lower waiting time for longer processes, throughput often increased. By dynamically adjusting the priority of jobs based on their response ratios, HRRN ensures that processes with higher urgency are executed promptly, thereby reducing overall response times and enhancing system performance [8].
* Disadvantage: One potential drawback is the increased computational overhead associated with dynamically calculating response ratios for each process, particularly in systems with a large number of concurrent jobs. This overhead may impact system performance and scalability. Additionally, the non-preemptive nature of HRRN may result in suboptimal resource utilization in situations where preemptive scheduling would be more advantageous, potentially leading to inefficient use of CPU resources [8].

## First Come First Serve

* Advantage: The processes would be dispatched according to their arrival time on the ready queue[5]. Another word means that the process will be executed based on their arrival time.
* Disadvantage: This schedule is not applicable in real-life systems, because the average time is often long and if one process takes too long then the shorter process has to wait for its turn.

# Methods/Algorithms

To represent the closest to the commercial CPU model, the processes are 75 for all of the methods. The average number of processes running on a normal computer is 70 to 80 processes [4][6]. The switch time is 5 microseconds.

1. *Round Robin*

For Round Robin, there will be five different quanta, these are 5, 10, 20, 40, and 80 milliseconds. These quantum were picked because of the most optimal and often used quantum based on another research [7][1]. The burst time is added together from the other research [7][1], and the average is 30 milliseconds, which is used for all of the burst times of processes in this simulation. The test ran three times to get the average of CPU utilization, throughput, turnaround time, waiting time, and response time. After having all of the data, highlight the first and second following the scheduled objective, then compare different quantum by which one has the most highlighted sections.

1. *Multilevel Feedback Queues*

The MFQ has level 1 is Round Robins, level 2 is also Round Robins, and the last level is FCFS. After comparing the results of Round Robin, the chosen Multi Feedback Queues are 5 milliseconds/5 milliseconds/FCFS, 10 milliseconds/10 milliseconds/FCFS, 20 milliseconds/20 milliseconds/FCFS, 40 milliseconds/40 milliseconds/FCFS. The test ran three times to get the average of CPU utilization, throughput, turnaround time, waiting time, and response time. After having all of the data, highlight the first and second following the schedule objective, then compare different MFQ by which one has the most highlight sections.

1. *Shortest Process Next*

For each simulation of the Shortest Process Next (SPN) scheduling algorithm, a set of 100 processes was generated, each with a randomly assigned burst time representing its processing requirement. These processes were then queued up for execution on the available 4 CPUs. Throughout each simulation run, the algorithm continuously selected the process with the shortest remaining burst time from the queue and allocated CPU resources to it until completion. The simulation tracked various performance metrics, including throughput, average turnaround time, average waiting time, and average response time across the 8 runs. By systematically varying the characteristics of the generated processes and observing the algorithm's behavior under different conditions, the simulations provided insights into SPN's efficiency in prioritizing shorter tasks and managing system resources effectively.

1. *Highest Response Ratio Next*

Similarly, for each simulation of the Highest Response Ratio Next (HRRN) scheduling algorithm, a pool of 100 processes was created with randomly assigned burst times. The algorithm dynamically calculated the response ratio for each process based on its waiting time and burst time. The process with the highest response ratio was selected for execution on the available CPUs, aiming to minimize response times and enhance system responsiveness. Through 8 simulation runs, the algorithm's performance was evaluated in terms of throughput, average turnaround time, average waiting time, and average response time. By assessing HRRN's ability to prioritize processes based on their urgency and dynamically adjust scheduling decisions, the simulations offered insights into its effectiveness in optimizing system performance and resource utilization.

# Results

1. Round Robins

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 5 ms | 10 ms | 20 ms | 40 ms | 80 ms |
| Throughput | 42.74 | 43.24 | 42.8 | 42.61 | 42.01 |
| CPU USAGE | 96.50 | 96.48 | 97.77 | 95.97 | 95.87 |
| Wait | 522.66 | 542 | 563.66 | 408.33 | 470 |
| Response | 23.66 | 53.66 | 81.33 | 119.66 | 134.66 |
| Turnaround | 672 | 690.66 | 716.66 | 558.66 | 623 |
| Time of Completion | 1741 | 1735.66 | 1754.33 | 1761 | 1786.33 |

5 milliseconds and 10 milliseconds quantum are the most effective based on the data for Round Robins.

1. Multilevel Feedback Queues

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5/5/FCFS | 10/10/FCFS | 20/20/FCFS | 40/40/FCFS |
| THROUGHPUT | 42.8 | 42.86 | 43.05 | 42.05 |
| CPU USAGE | 94.65 | 96.26 | 95.29 | 95.41 |
| WAIT | 413.33 | 531.66 | 494.33 | 501.66 |
| RESPONSE | 37.66 | 57.66 | 71 | 141.66 |
| TURNAROUND | 560.33 | 682 | 641.33 | 653.33 |
| TIME OF COMPLETION | 1754 | 1750.33 | 1742 | 1787 |

20 millisecond/20 millisecond/FCFS is the most effective based on the data for Multilevel Feedback Queues.

1. Shortest Process Next

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Throughput** | **Average Turnaround** | **Average Waiting** | **Average Reponse** |
|  | 72.36 | 69.23 | 4.43 | 1.14 |
|  | 79.68 | 77.06 | 6.26 | 0.99 |
|  | 76.22 | 67.29 | 5.79 | 1.32 |
|  | 81.7 | 68.49 | 4.89 | 1.13 |
|  | 72.99 | 70.19 | 3.59 | 0.79 |
|  | 74.91 | 73.73 | 5.63 | 1.2 |
|  | 72.57 | 79.92 | 6.12 | 1.04 |
|  | 74.74 | 70.54 | 5.44 | 0.98 |
| **Average** | **75.65** | **72.06** | **5.27** | **1.07** |

1. Highest Response Ratio Next

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Throughput** | **Turnaround** | **Waiting** | **Reponse** |
|  | | 86.43 | 70.96 | 5.56 | 1.98 |
|  | | 75.3 | 81.59 | 6.89 | 2.9 |
|  | | 75.53 | 84.97 | 6.97 | 2.34 |
|  | | 76.51 | 81.76 | 6.46 | 2.39 |
|  | | 72.89 | 86.47 | 7.57 | 3.01 |
|  | | 78.19 | 82.55 | 6.95 | 2.88 |
|  | | 80.13 | 86.07 | 7.17 | 1.94 |
|  | | 83.4 | 79.58 | 6.68 | 2.15 |
| **Average** | | **78.55** | **81.74** | **6.78** | **2.45** |

# Conclusion

In conclusion, our study highlights the critical importance of efficient process scheduling in commercial computer systems, where timely response to user demands is paramount. Through meticulous analysis and experimentation with discrete event simulation, we have identified key insights into optimizing process scheduling algorithms to enhance system performance and user experience. The determination of optimal time quantums for Round Robin and Multilevel Feedback Queue algorithms provides practical guidance for system administrators and developers in real-world implementations. Moreover, our findings regarding the prioritization of response time highlight the nuanced trade-offs between different scheduling strategies, with Shortest Process Next and Highest Response Ratio Next emerging as promising candidates. Notably, the observation of reduced starvation with Highest Response Ratio Next shows its potential for mitigating fairness concerns in process scheduling. Overall, our research contributes to the growing body of knowledge on efficient resource management in commercial computing environments and provides valuable insights for further advancements in this critical area of study.

##### References

1. “A Priority based.” Available: https://www.idc-online.com/technical\_references/pdfs/information\_technology/A%20Priority%20based.pdf
2. M. Y. Shakor, “Scheduling and Synchronization Algorithms in Operating System: A Survey,” Journal of Studies in Science and Engineering, vol. 1, no. 2, pp. 1–16, Nov. 2021, doi: https://doi.org/10.53898/josse2021121.
3. “Operating Systems I -4’th Stage-Lecture 6 Lecturer: Hawraa Shareef.” Available: https://www.uobabylon.edu.iq/eprints/publication\_11\_2849\_1410.pdf
4. “Pushing the Limits of Windows: Processes and Threads,” TECHCOMMUNITY.MICROSOFT.COM, Jun. 27, 2019. https://techcommunity.microsoft.com/t5/windows-blog-archive/pushing-the-limits-of-windows-processes-and-threads/ba-p/723824
5. R. J. Matarneh, “Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes,” American Journal of Applied Sciences, vol. 6, no. 10, pp. 1831–1837, Oct. 2009, doi: https://doi.org/10.3844/ajassp.2009.1831.1837.
6. R. Ware, “9 Windows Processes You Can End Safely to Improve Performance,” MUO, Sep. 29, 2022. https://www.makeuseof.com/windows-processes-end-safely-performance/#:~:text=Processes%20are%20basically%20programs%20being (accessed May 14, 2024).
7. S. Zouaoui, L. Boussaid, and A. Mtibaa, “Priority based round robin (PBRR) CPU scheduling algorithm,” International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 1, p. 190, Feb. 2019, doi: <https://doi.org/10.11591/ijece.v9i1.pp190-202>.
8. Behera, H. S., Swain, B. K., Parida, A. K., & Sahu, G. (2012). A new proposed round robin with highest response ratio next (rrhrrn) scheduling algorithm for soft real time systems. International Journal of Engineering and Advanced Technology, 37, 200-206.
9. Behera, H. S., Naik, R. K., & Parida, S. (2012). Improved multilevel feedback queue scheduling using dynamic time quantum and its performance analysis. International Journal of Computer Science and Information Technologies, 3(2), 3801-3807.
10. Hashim Yosuf, R., Mokhtar, R. A., Saeed, R. A., Alhumyani, H., & Abdel-Khalek, S. (2022). Scheduling Algorithm for Grid Computing Using Shortest Job First with Time Quantum. Intelligent Automation & Soft Computing, 31(1).

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**