Multicore CPU Process Scheduling Simulator

Problem Statement

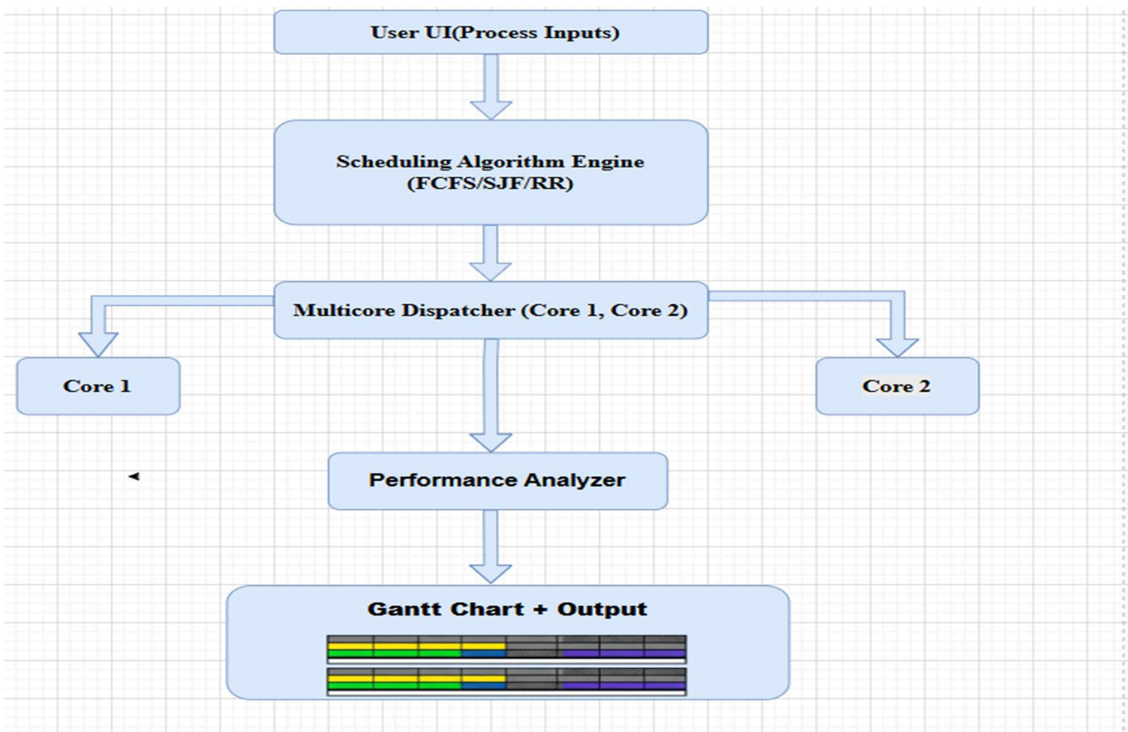
Modern computers are all equipped with multicore processors, but it is hard for students to understand how process execution could be scheduled over multiple cores. Current OS schedulers are typically background tasks that can't be seen. This project implements a simulator of how the processes are scheduled on multi-core system using different CPU scheduling algorithms, as well as analyse the performance measures such as waiting time; turn-around time and CPU utilization.

Abstract

CPU Scheduling is an important aspect in determining the efficiency, responsiveness, and performance of modern computing systems, especially in the advent of multicore processors. While there are several IEEE research papers that have proposed simulators, models, and comparisons of CPU Scheduling algorithms such as Round Robin, Priority, or Shortest Job First, the existing solutions are generally plagued by the assumption of single-core processors, the lack of visualization of real-time execution, or the lack of a comprehensive comparison framework. Moreover, important metrics such as response time, throughput, and CPU utilization are generally incompletely addressed or analysed. To overcome these shortcomings, this paper proposes a Multicore CPU Process Scheduling Simulator that models real-time process scheduling on multiple CPU cores and offers an interactive comparison platform for popular process scheduling algorithms such as FCFS, SJF, Priority Scheduling, and Round Robin. In contrast to existing solutions like dynamic Round Robin models and survey-based analytical studies [1] – [10], the proposed system combines real-time response time calculation, interactive multicore Gantt chart display, and efficiency-focused performance analysis into a single GUI-based environment. The simulator simulates the execution of processes in a time-driven fashion using arrival time, burst time, and priority parameters, which closely resemble the actual operating system process scheduling. Real-time execution simulation enables the simulation of parallelism, pre-emption, and core-level workload distribution, which are not actually demonstrated in many simulators. Unlike the previous comparative and grid-based scheduling simulation studies, the proposed framework calculates Waiting Time, Turnaround Time, Response Time, Throughput, and CPU Utilization in a uniform fashion for various multicore configurations, which allows for a direct comparison of efficiency. The originality of this work is found in its real-time multicore-aware scheduling visualization, comparative analysis, and complete practical implementation, filling the gap between theoretical scheduling research and practical operating

system behaviour. Through the extension and improvement of ideas presented in previous IEEE papers by incorporating real-time capture and efficiency analysis, the proposed simulator can be considered both an educational tool and a framework for understanding modern multicore CPU scheduling systems.

High-Level Architectural Diagram



References

1. **A.A. Alsulami, Q. A. Al-Haija, M. I. Thanoon, and Q. Mao**,"Performance Evaluation of Dynamic Round Robin Algorithms for CPU Scheduling,"*IEEE SoutheastCon*, Huntsville, AL, USA, 2019, pp. 1–5.

2. **K. Vayadande, P. Sheth, D. Pawal, A. Pathak, K. Paralkar, and S. Patil**,"Simulation of CPU Scheduling Algorithms for Efficient Execution of Processes,"*2023 International Conference for Advancement in Technology (ICONAT)*, Goa, India, 2023, pp. 1–6.

3. **S. Sharma, A. Chhabra, and S. Sharma**,"Comparative Analysis of Scheduling Algorithms for Grid Computing,"*2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, 2015, pp. 349–354.

4. **S. Zouaoui, L. Boussaid, and A. Mtibaa**,"CPU Scheduling Algorithms: Case and Comparative Study,"*2016 International Conference on Sciences*

and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 2016, pp. 158–164.

5. **1.R. Jaiswal**, "Comparison of CPU Scheduling Algorithms," *IEEE International Conference on Computing, Communication and Automation (ICCCA)*, 2024, pp. 1–5.
6. **H. Singh and A. Kaur**, "Performance Analysis of CPU Scheduling Algorithms Using Simulation," *IEEE International Conference on Computing for Sustainable Global Development (INDIA Com)*, 2018, pp. 2456–2460.
7. **M. A. Khan, M. Rashid, and A. Khan**, "Comparative Study of CPU Scheduling Algorithms," *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, 2019, pp. 1–6.
8. **P. Kumar and R. Kumar**, "Analysis of Multicore CPU Scheduling Algorithms," *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 1736–1740.
9. **S. S. Patil and R. S. Patil**, "Performance Evaluation of Multicore Scheduling Algorithms," *IEEE International Conference on Advances in Computer Engineering and Applications (ICACEA)*, 2015, pp. 249–253.
10. **A. Gupta and S. Jain**, "A Survey on CPU Scheduling Algorithms," *IEEE International Conference on Computational Intelligence and Communication Networks (CICN)*, 2016, pp. 490–495.
11. **X. Yao, P. Geng and X. Du,** "A Task Scheduling Algorithm for Multi-core Processors," *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies*, Taipei, Taiwan, 2013, pp. 259-264.