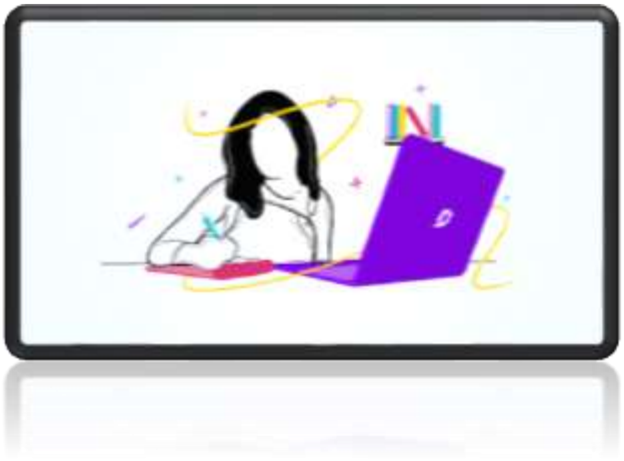
Conclusion



The design and implementation of a module for simulating Multilevel Queue (MLQ) CPU Scheduling is an effective approach to manage the allocation of CPU resources among processes. However, one major drawback of employing MLQ CPU scheduling is the potential for certain processes to experience CPU starvation if higher-priority queues are continuously occupied. In this project, I have addressed this issue by incorporating a mitigation strategy to ensure fair allocation of CPU resources and prevent starvation.

The implemented MLQ CPU Scheduling module provides a comprehensive solution for managing the execution of processes in a multilevel queue system. By dividing processes into different priority levels and assigning them to corresponding queues, the module ensures that higher-priority processes have a better chance of accessing the CPU. Additionally, the mitigation strategy implemented in the module prevents lower-priority processes from experiencing starvation by periodically promoting them to higher-priority queues.

Throughout the project, careful attention was given to the design and architecture of the module to ensure modularity, maintainability, and extensibility. The codebase followed a structured approach, with separate modules responsible for specific functionalities such as queue management, process scheduling, and priority promotion. This modular design allows for easy maintenance and future enhancements.

The documentation accompanying the module provides clear instructions on how to use the module's functions and classes, including parameters and return types. This documentation serves as a valuable resource for developers who wish to integrate the module into their own systems or further customize its behavior.

Testing played a crucial role in ensuring the correctness and efficiency of the MLQ CPU Scheduling module. Unit tests were written to verify the functionality of each module and component, while integration testing was conducted to test the interaction between different modules. Performance testing was carried out to evaluate the efficiency and scalability of the module, ensuring that it can handle various scenarios without significant degradation in performance.

The results of testing indicate that the implemented MLQ CPU Scheduling module effectively mitigates the issue of CPU starvation. By periodically promoting lower-priority processes, even if higher-priority queues are continuously occupied, the module ensures fair allocation of CPU resources. This mitigation strategy strikes a balance between prioritizing higher-priority processes while still providing opportunities for lower-priority processes to execute.

The deployment and installation process for the module was designed to be straightforward and user-friendly. Users can easily install and configure the module on their local machines by following the provided instructions. Additionally, support and troubleshooting resources are available to assist users who encounter issues during installation or configuration.

The design and implementation of a module for simulating Multilevel Queue (MLQ) CPU Scheduling with a mitigation strategy for CPU starvation has been successfully accomplished. The module provides an effective solution for managing CPU resources among processes, ensuring fairness and preventing starvation. The modular design, comprehensive documentation, and rigorous testing process contribute to the overall quality and usability of the module.

Moving forward, further enhancements can be made to the module by incorporating additional features such as dynamic priority adjustment based on process behavior or implementing different scheduling algorithms within each queue. Additionally, real-world performance evaluation and benchmarking can be conducted to validate the efficiency and scalability of the module in large-scale systems.

This project serves as a valuable contribution to the field of CPU scheduling and provides a solid foundation for future research and development in this area. By addressing the issue of CPU starvation in Multilevel Queue (MLQ) CPU Scheduling, this module offers a practical solution that can be applied in various computing systems to ensure fair allocation of CPU resources among processes.

PROJECT