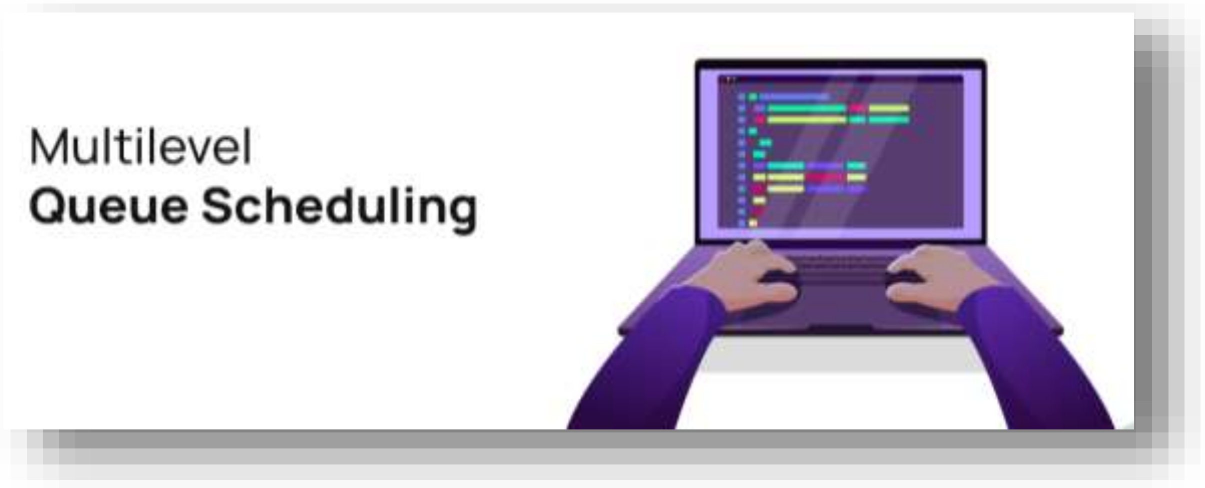
Introduction

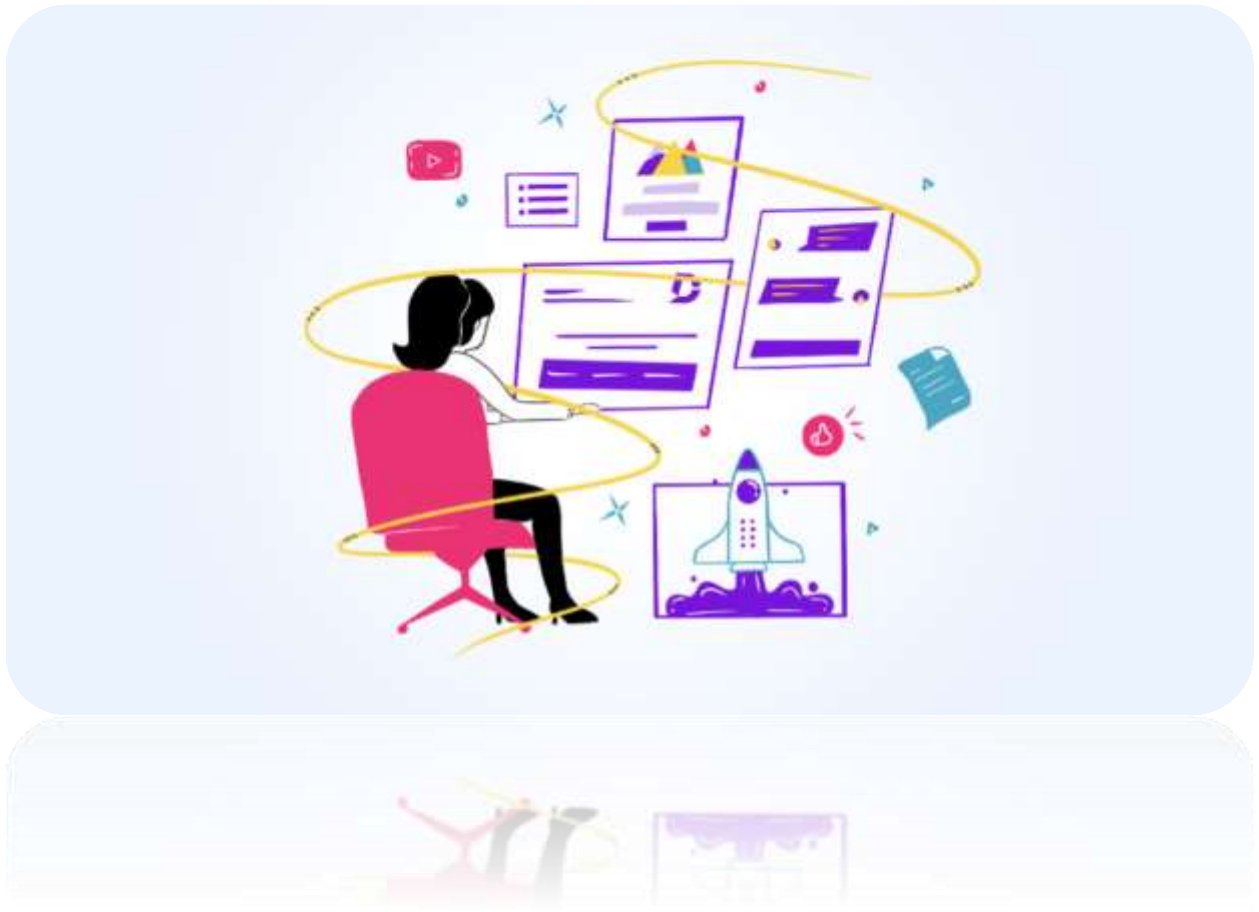


Multilevel Queue (MLQ) CPU scheduling is a widely used technique in operating systems. It provides a way to prioritize processes based on their characteristics and requirements. In this project, I will design and implement a module for simulating MLQ CPU scheduling. One of the main challenges in MLQ scheduling is the possibility of CPU starvation for lower priority processes, which can occur if higher-priority queues are continuously occupied. Therefore, my implementation will include a mechanism to mitigate this issue and ensure that all processes get a fair share of CPU time. I will explore different strategies to achieve this goal and evaluate their performance through simulations. This project will provide a hands-on experience in designing and implementing a critical component of modern operating systems and will help enhance my understanding of CPU scheduling algorithms.

The Multilevel Queue (MLQ) CPU Scheduling is a technique used to manage the allocation of CPU time to processes based on their priority levels. It is a widely used technique in modern operating systems due to its ability to provide better performance and fairness in the allocation of CPU time. However, one of the major drawbacks of employing MLQ CPU scheduling is the potential for certain processes to experience CPU starvation if higher-priority queues are continuously occupied. In such cases, lower-priority processes may not receive adequate CPU time, leading to a decrease in their overall performance and response time.

To address this issue, I aim to design and implement a module for simulating MLQ CPU Scheduling that possesses the capability to mitigate the problem of CPU starvation. My implementation will include a mechanism that ensures fairness in the allocation of CPU time to processes while preventing starvation. The purpose of this project is to provide a solution that can effectively manage the allocation of CPU time to processes in a way that ensures fairness and prevents starvation. My implementation will take into account the priority levels of processes and allocate CPU time accordingly.

The module will be designed to simulate the behavior of an operating system scheduler and will be implemented using programming languages, data structures, and algorithms that are suited for this purpose. The module will also include a user interface that enables users to interact with the simulation and view the results. This report will provide an overview of the project goals, requirements, design, implementation, and testing results. I will also discuss the potential for future enhancements and improvements to the module.



This project aims to design and implement a module for simulating MLQ CPU Scheduling that addresses the problem of CPU starvation for certain processes due to continuously occupied higher-priority queues. The module will ensure fairness in the allocation of CPU time to processes while preventing starvation, and will be implemented using programming languages, data structures, and algorithms that are suited for this purpose.