Case Study ID: -2320030120

MULTILEVEL FEEDBACK QUEUE SCHEDULING

Introduction

Overview

Multilevel Feedback Queue (MLFQ) scheduling is an advanced CPU scheduling algorithm designed to optimize process execution time and responsiveness in operating systems. It employs multiple queues with varying priority levels, allowing processes to move between queues based on their behaviour and requirements.

Objective

The primary objective of this report is to analyse the MLFQ scheduling algorithm, its implementation, benefits, challenges, and how it can be effectively integrated into modern operating systems to enhance performance.

Background

Organization/System Description

MLFQ scheduling is commonly implemented in operating systems to manage process scheduling efficiently. It is designed to provide a balance between responsiveness for interactive tasks and throughput for batch tasks.

Current Network Setup

The current network setup involves a traditional round-robin or first-come-first-served scheduling, which may lead to inefficiencies in handling varying process priorities and execution times.

Problem Statement

Challenges Faced

- Inefficient CPU utilization due to static scheduling algorithms.
- Increased waiting times for lower-priority processes.
- Difficulty in adapting to the dynamic nature of process workloads.

Proposed Solutions

Approach

Implementing the MLFQ scheduling algorithm involves creating multiple queues with different priority levels, allowing processes to be dynamically adjusted based on their CPU burst time and I/O operations.

Technologies/Protocols Used

- Operating System Kernels (Linux, Windows)
- Scheduling Algorithms
- Performance Monitoring Tools

Implementation

Process

1. Design the MLFQ structure with multiple queues.
2. Establish rules for process promotion and demotion between queues.
3. Implement the scheduling algorithm in the OS kernel.

Implementation

- Use of system calls to manage process states.

- Integration with existing scheduling mechanisms.

Results and Analysis

Outcomes

- Improved CPU utilization and reduced average waiting time.
- Enhanced responsiveness for interactive applications.

Analysis

Data collected from performance metrics will be analysed to compare MLFQ against traditional scheduling methods, focusing on throughput, turnaround time, and response time.

Security Integration

Security Measures

- Implement access controls to prevent unauthorized manipulation of scheduling parameters.
- Regular audits and monitoring to detect anomalies in process scheduling.

Conclusion

Summary

The MLFQ scheduling algorithm presents a robust solution to the challenges faced by traditional scheduling methods, offering improved performance and adaptability.

References

- Geeks for Geeks
- Tutorials Point
- Course era

NAME: V. Charan Sai

ID-NUMBER: 2320030120

SECTION-NO: 1

31/8.