

Smart Task Scheduler with Priority Queue

Introduction

A Smart Task Scheduler enhanced with a Priority Queue is a system developed to efficiently manage tasks based on their urgency or importance. Unlike traditional queue-based schedulers that operate on a First-In-First-Out (FIFO) principle, a priority queue ensures that high-priority tasks are executed first, regardless of when they were added. This dynamic scheduling mechanism significantly improves responsiveness and resource management. Such schedulers are widely used in real-time applications, including operating systems, healthcare, finance, and automated industrial systems, where timely task execution is critical. By integrating a smart algorithm that sorts and manages tasks with different priority levels, the system enhances productivity, time management, and reliability.

Abstract

This paper introduces the concept of a Smart Task Scheduler that leverages a priority queue to optimize task management and execution. Unlike traditional scheduling methods, this approach dynamically prioritizes tasks based on predefined criteria such as urgency, deadlines, or importance. By utilizing the priority queue data structure, the scheduler ensures that high-priority tasks are processed before lower-priority ones, regardless of their arrival order.

This research explores the advantages and practical applications of the system, emphasizing its ability to enhance responsiveness, minimize delays, and optimize resource utilization, especially in real-time environments. Furthermore, various implementation strategies for priority queues are examined, along with their impact on system performance. The paper provides insights into designing a robust, intelligent, and adaptable task scheduling system capable of managing diverse and dynamic workloads efficiently.

Tools used

- Programming Language: Java
- GUI Framework: Java Swing
- Storing task data: File I/O (tasks.txt)
- Timer/Reminders: Java Timer, TimerTask
- IDE: Visual Studio Code

Steps Involved in Building the Project

1. Design of Task Structure:

- Created a Task class with fields like title, priority, due date.
- Implemented Comparable interface to define natural ordering by priority.

2. GUI Development:

- Built user interfaces for:
 - Adding a new task
 - Viewing all tasks (in order of priority)
 - Editing or deleting tasks
 - Filtering (e.g., today's tasks, high priority)
- Used JPanel, JButton, JTable, JTextField, JComboBox, and other Swing components.

3. Priority Queue:

- Used Java's PriorityQueue to store tasks dynamically based on priority.
- Ensured that the queue reorders automatically as tasks are added or removed.

4. Task Management Features:

- Enabled Add, Edit, Delete, and View functionalities.
- Implemented filters to view tasks by priority or due date.

5. Reminder System Implementation:

- Used `java.util.Timer` and `TimerTask` to periodically check for due tasks and display alerts or console reminders.

6. Data Persistence:

- Implemented file I/O (e.g., tasks.txt) to save and load tasks on startup/shutdown.
- Ensured the system preserves task data across sessions.

7. Deployment

- Compiled the project into a .jar executable using javac and jar tools.
- Created a proper manifest.txt with Main-Class specified for execution.

Conclusion

The Smart Task Scheduler with Priority Queue provides an effective solution for managing tasks based on urgency and importance. By prioritizing tasks dynamically instead of using traditional FIFO methods, the system improves time management and responsiveness.

Using Java and Swing for the interface, and a priority queue for task handling, the project demonstrates how intelligent scheduling can enhance productivity in both personal and professional contexts. Future improvements could include database integration, reminders, or cloud-based task sharing.