Wipro NGA Program -C++ LSP batch

Capstone Project

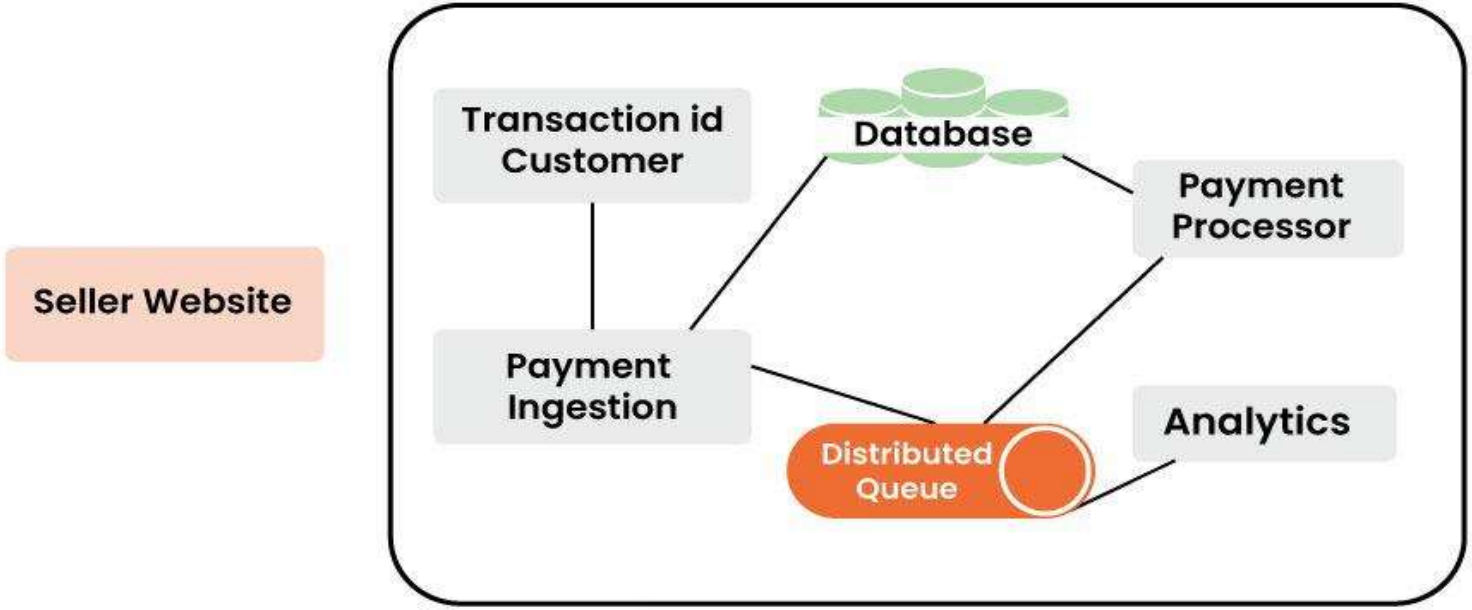
Project title - Multi threaded Banking System
Presented by - Dharani Palagiri

Project Overview

A multithreaded banking system aims to efficiently handle concurrent transactions, improving performance, scalability, and user experience. This project will involve designing and implementing a banking system using Linux and multithreading concepts.

In this project we will learn how to interact with multiple tasks happening simultaneously.

System Design



Introduction

A multithreaded banking system is a software application designed to simulate real-world banking operations, allowing multiple users to perform transactions concurrently. Built upon the Linux operating system, it leverages the power of multi-threading to efficiently handle simultaneous requests and improve system performance.

Motivation

A multithreaded banking system in Linux is motivated by the need to handle a large number of concurrent transactions efficiently and responsively. Here's a breakdown of the key motivations:

Multiple threads can handle multiple transactions simultaneously, significantly improving the system's capacity to process requests.
Clients don't have to wait for long periods for their transactions to complete, enhancing user experience.
The system can handle increasing numbers of concurrent users by adding more threads, without requiring additional hardware.

Project Scope

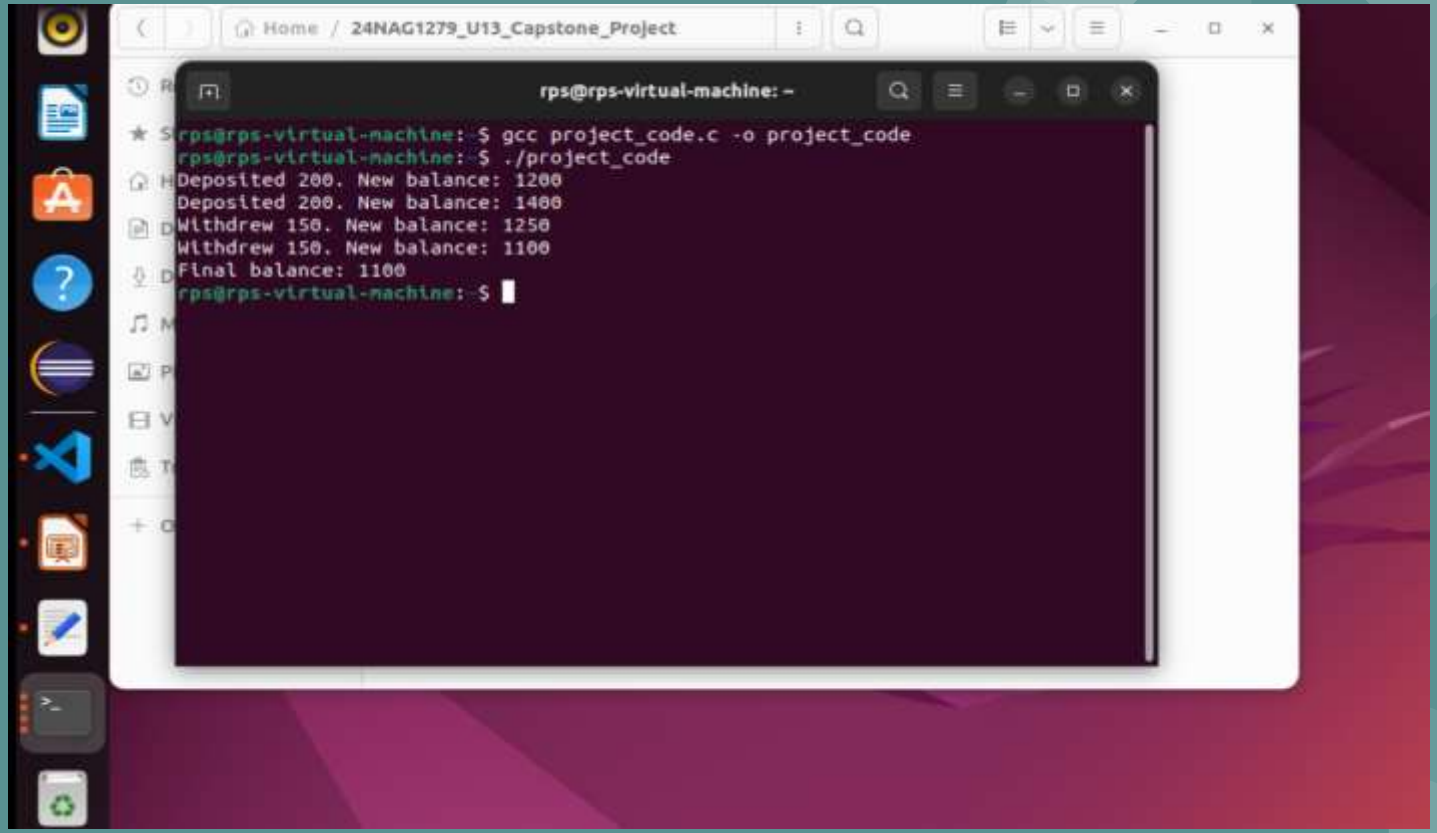
The scope of a multithreaded banking system can vary widely based on the specific requirements and target audience.

By carefully considering these aspects, you can define the scope of your multithreaded banking system to meet the specific needs of your target market and business objectives.

List of functions

Banking Functions:
-> Account Management
-> Transaction Management
-> User Management
-> Security
-> Thread Management

Output


Home / 24NAG1279_U13_Capstone_Project
rps@rps-virtual-machine: ~
rps@rps-virtual-machine: $ gcc project_code.c -o project_code
rps@rps-virtual-machine: $ ./project_code
Deposited 200. New balance: 1200
Deposited 200. New balance: 1400
Withdrew 150. New balance: 1250
Withdrew 150. New balance: 1100
Final balance: 1100
rps@rps-virtual-machine: $

Future Amendments

A multithreaded banking system, while robust, requires continuous evolution to adapt to changing technological landscapes, regulatory requirements, and customer expectations.

By incorporating these future amendments, a multithreaded banking system can remain competitive, secure, and customer-centric in an ever-changing technological landscape.

Addressing Common Challenges

1. Data Consistency
2. Concurrency Control
3. Security
4. Performance Optimization
5. Database Selection
6. Network Communication

Conclusion

Multithreaded banking systems offer a substantial advantage in handling the high concurrency demands of the financial industry. By enabling simultaneous processing of multiple transactions, these systems significantly enhance performance, scalability, and responsiveness.