Software Engineering Concepts

SE - Lab 04

Topic: Concept of Multithreading
Estimation: 15 hours.

Lab Overview

The purpose of this lab is to deepen your understanding of multithreading and concurrency in Java through practical implementation of various concepts.

Task 1: File Processing with Multithreading

Create a Java program that reads multiple text files concurrently using threads. Each thread should:

- Read a different text file
- Count the number of words, lines, and characters in the file
- Print the results with thread name and processing time
- Use at least 3 different files and 3 threads

Requirements:

- Implement using both Thread class extension and Runnable interface
- Handle file not found exceptions properly
- Measure and display execution time for each thread

Task 2: Producer-Consumer with Priority Queue

Implement a Producer-Consumer pattern with the following specifications:

- Use a priority queue as the shared buffer (capacity: 10)
- Producer generates random numbers (1-100) with random priorities
- Consumer processes items based on priority (highest first)
- Create 2 producer threads and 3 consumer threads
- Run for 30 seconds and display statistics

Requirements:

- Implement proper synchronization
- Display buffer state after each operation
- Show total items produced and consumed by each thread

Task 3: Thread Pool Web Crawler Simulation

Create a web crawler simulation using thread pools:

- Simulate downloading web pages (use Thread.sleep() for delay)
- Each "page" has a random download time (1-5 seconds)
- Use different thread pool types: FixedThreadPool, CachedThreadPool, SingleThreadExecutor
- Process 20 URLs and compare performance

Requirements:

- Measure total execution time for each thread pool type
- Handle timeouts (max 10 seconds per page)
- Display which thread processed each URL

Task 4: Bank Account Race Condition Demonstration

Create a program that demonstrates race conditions and their solutions:

- Implement a BankAccount class with deposit/withdraw methods
- Create both synchronized and non-synchronized versions
- Run 10 threads performing 100 random transactions each
- Compare final balances between synchronized and non-synchronized versions

Requirements:

- Start with initial balance of $1000
- Each transaction should be $1-50 (random)
- Display transaction history and final balance
- Explain the difference in results

Submission Guidelines:

- Create separate Java files for each task
- Include comments explaining your approach
- Add README.md with execution instructions
- Submit all source code and output screenshots
- Deadline: One week from assignment date