

Project Title:

**[The Search-Insert-Delete Problem]**

Submitted by:

**[Urwah Imran] [2021-CE-15]**

**[Samreen Razzaq] [2021-CE-13]**

**[Ayesha] [2021-CE-18]**

**[Aiman Malik] [2021-CE-23]**

Submitted to:

**[Mam Darakhshan Abdul Ghaffar]**

Course:

**[CMPE-331L: Operating Systems]**

Semester:

**[5th]**

Date of Submission:

**[10-Nov-23]**

**Department of Computer Engineering**

**University of Engineering and Technology, Lahore**

Table of Contents

[Abstract 2](#_Toc149923044)

[1. Introduction 3](#_Toc149923045)

[2. Problem statement 3](#_Toc149923046)

[3. Proposed Methodology 3](#_Toc149923047)

[4. Expected outcome/results 3](#_Toc149923048)

[5. Applications in Operating systems 3](#_Toc149923049)

[References 3](#_Toc149923050)

# 

# 

# Abstract

The “Search-Delete-Insert-Problem”, a concurrency challenge in operating systems. In this problem, three types of threads, namely searchers, inserters, and deleters, share access to a singly linked list. Searchers examine the list concurrently, while inserters add new items to the list, and deleters remove items from anywhere in the list. However, these operations must be categorized into three-way mutual exclusion, where searchers can run concurrently, insertions require mutual exclusion, and deletion should be mutually exclusive with searches and insertions. The report outlines the problem, proposes a methodology to address it, anticipates the expected outcomes, discusses potential applications in operating systems, and provides references for further exploration.

# Introduction

Concurrency in operating systems poses unique challenges, and the "Search-Insert-Delete Problem" exemplifies one such challenge. In this problem, multiple threads with different roles need to access a singly-linked list while adhering to strict rules of mutual exclusion. This report explores this problem in depth and proposes a methodology to tackle it effectively.

# Problem Statement

The problem entails designing a concurrent system in which searchers, inserters, and deleters can access a singly-linked list. Searchers operate concurrently without interfering with each other, inserters must ensure mutual exclusion to prevent conflicts, and deleters need exclusive access, excluding searches and insertions. The task is to create a system that enforces this three-way categorical mutual exclusion effectively.

# Proposed Methodology

To address this problem, a thread synchronization approach will be employed. The searchers will execute concurrently, as they merely examine the list. Insertions will be guarded by mutual exclusion mechanisms to avoid simultaneous insertions, and deletions will require exclusive access. Mutexes, semaphores, or other synchronization primitives will be employed to ensure the proper synchronization of these operations.

# Expected Outcome/Results

The expected outcome is a functional concurrent system that enforces the three-way categorical mutual exclusion. Searchers should be able to operate concurrently, inserters must execute with mutual exclusion, and deleters should have exclusive access. The system should prevent conflicts and ensure the correct execution of these operations.

1. **Applications in Operating System**

The concept of three-way categorical mutual exclusion is not limited to this specific problem but finds applications in various aspects of operating systems. It can be used in resource management, file system operations, **and ensuring** thread

safety in multi-threaded applications. The principles learned from solving this problem can be applied to design efficient and robust concurrent systems.

# References

<https://www.geeksforgeeks.org/search-insert-and-delete-in-a-sorted-array/>

<https://youtu.be/1KyMN3IB4BE?si=Jeks4udTkamWKRbW>