UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS 226 - Concurrent Systems

Practical 6

- **Date issued:** 23 October 2023
- **Deadline:** 29 October 2023 (Midnight)

1 Introduction

1.1 Objectives and Outcomes

This practical aims to further explore synchronization by evaluating locking in alternative data structures to queues.

You must complete this assignment individually. Copying will not be tolerated.

1.2 Submission and Demo Bookings

You are NOT provided with any skeleton code for this practical, you will have to implement everything yourself.

Submit your code to **clickup** before the deadline.

You will have to demonstrate each task of this practical during the **physical** practical lab session. So be sure to create copies of your source code for each task separately. Booking slots will be made available for the practical demo.

1.3 Mark Allocation

For each task in this practical, in order to achieve any marks, the following must hold:

- Your code must produce console output. (As this is not marked by fitchfork, formatting is not that strict)
- Your code must not contain any errors. (No exceptions must be thrown)

- Your code may not use any external libraries apart from those highlighted in the textbook.
- You must be able to explain your code to a tutor and answer any questions asked.

The mark allocation is as follows:

Task Number	Marks
Task 1	5
Task 2	5
Total	10

2 Practical Requirements

You are tasked in simulating a security protocol that uses a central database to schedule jobs on the system. There are two user types in the system with different authorisation level:

- Developer
  - Responsible for scheduling jobs that need to be done
  - The job requests are inserted into a central database.
  - Each job has the number of hours required to preform the job, generated randomly between 1-24.
- System Administrator
  - Inspect the job schedule by developers and approves/disapprove.
  - The job requests are removed from the central database.
  - A job with hours less than a randomly generated number between 1-24 is accepted.

Note

- Your code must demonstrate concurrency using a variable number of threads, at a ratio 1:2 System Admin to Developer, ideally 2:4.
- Each developer must schedule at-least 3 jobs.

Output
The following output is expected:

1. Demonstrate insertion: (IN) [thread-name] [job-number] [hours]
2. Demonstrate removal: (OUT) [thread-name] [job-number] [hours] [approval-status]

2.1 Tasks 1 - Lock-free Queue

For this task you must use an **unbounded lock-free queue** as the central database

2.2 Task 2 - Lock-free Stack

For this task you must use a **lock-free stack** as the central database