Queen's School of Computing
CISC324 – Fall 2024
Instructor: Dr. Anwar Hossain
Lab 3 - Due Date: Nov. 1, 2024

Lab 3: Synchronization and Deadlock Avoidance

Objective

This lab will introduce students to the use of semaphores to handle synchronization and explore how deadlocks can occur and be managed. Students will apply synchronization concepts to solve a practical problem.

Scenario: Writers and Editors Working on Articles

In this lab, multiple **writers** and **editors** collaborate to publish articles. Each writer drafts an article and submits it for review by an editor. However, there are **limited editors** and **limited slots** for **articles under review** at the same time.

- Writers must **acquire a slot** to submit their article and **wait for an editor to review it**.
- You need to ensure **no two editors** review the same article at the same time.
- You must ensure **proper synchronization** between the writers and editors using semaphores to avoid race conditions.
- The boiler plate code is given and there are TODO's in the code that you need to complete. The boilerplate code at this time only shows the messages, but no synchronization is implemented.
- Your task is the following:
    - Complete the code according to the TODO's. The code should simulate the stated deadlock-free scenario if implementation is done properly. Run the code and report the output in **lab3<last 4-digit student ID>.pdf**.
    - Comment out the code for TODO-1, run the code, and report the output and briefly explain what changes you notice in the same file (**NOTE**: **you might need to change the number of writers and/or number of editors in order to see some changes**)
    - Comment out the code for TODO-7, run the code, and report the output and briefly explain what changes you notice (put in the same file)

Queen's School of Computing
CISC324 – Fall 2024
Instructor: Dr. Anwar Hossain
Lab 3 - Due Date: Nov. 1, 2024

Prepare your environment!

The boilerplate code, provided to you, can work on any environment (Linux, Windows, MacOS, etc.) so please feel free to search for how to install Python on your machine.

IDEs that you can use (feel free to use any other IDE)

1. Visual Studio Community Edition for Windows-based machines only (it's something different from VS Code).
2. Visual Studio Code (Can work on any operating system).
3. PyCharm (Can work on any operating system).

Install Python on Your Machine

There are too many methods to install Python (feel free to use any other method):

1. Download and install Anaconda (which works on any operating system).
2. Download and install Python from the official website (works on any operating system).

Download the Boilerplate Code

There are two methods to download the boilerplate code:

1. Download the whole repository of the course (Zip file) and open the folder named "**Lab 3**".
2. If you are familiar with Git and Git commands, you can clone the repository or you can update your local repository if you already have it cloned previously.

What to submit?

1. Place all your source codes in a folder named in the following format:

   324-1234-Lab3, where 1234 stands for the last 4 digits of your student ID, e.g.: For student with ID 20196072, the folder should be named 324-6072-Lab3.

2. Place the file **lab3<last 4-digit student ID>.pdf** you created into the same folder above.

3. Compress the above folder using Zip (the extension must be .zip or .rar).
4. Log into OnQ, locate the lab's dropbox in OnQ, and upload the compressed folder.