COMP 2710: Project 5
Points Possible: 100
Deadline: **11:59pm Friday Apr. 22nd, 2022 (U.S. Central Time)**

There should be no collaboration among students. A student should not share any project code with any other student. Collaborations among students in any form will be treated as a serious violation of the University's academic integrity code.

Objectives*:*
1. Complete the source code to simulate producer/consumer problem.
2. Understand the basics of the POSIX thread library.
3. Build a binary program on a Linux machine.
4. Run the program and record the result of the simulation.

Requirements*:*
- Each student should **independently** accomplish this project assignment. You may discuss with other students to solve the coding problems.
- Important! Read and follow the I/O format specification carefully. We are using an auto-grade script to grade this project. It is very important for your final grade of this project! Any submission violating the format may not be correctly recognized.
- You are highly recommended to use a Linux operating system, because "pthread.h" is a header for the Unix/Linux (POSIX) API for threads.

1. Introduction to producer and consumer model
The producer and consumer model is to schedule how concurrent processes and threads access the resources. It contains:
1. **Producer**: One or multiple processes/threads that produce data or release hardware resource.
2. **Consumer**: The one process/thread that takes in data or uses hardware resource to do computations.

A producer could also be relatively a consumer to the output of another producer and vice versa.
3. **Buffer**: The destination to store the output from producers or resources and later accessed by another consumer.

Other concepts involved in our project:
4. **POSIX thread**: Threads mechanism that satisfy POSIX standard (most operating system).
5. **Mutex**: A "lock" that guarantee that only one person has the access.

In this project we use POSIX threads. The "pthread" is a POSIX thread library written in C++ and provides the basic functions.

To simplify our simulation, we assume there are only 2 POSIX threads. One is the consumer, the other is the producer. The producer generates 1 unit data each time to the buffer, and the

consumer takes 1 unit data from the buffer each time. The size of the buffer is 1. One unit data is just one integer. The producer generates integers 7, 14, 21 .... into the buffer and consumer read them out from the buffer.

2. Follow the Format Specification (10 Points)
In the source file "**project5_LastName_UserID.cpp**", you will find first four comment lines:

```
// #0#BEGIN# DO NOT MODIFY THIS COMMENT LINE!
// Firstname
// Lastname
// #0#END# DO NOT MODIFY THIS COMMENT LINE!
```

Your first task is modifying the two lines **between** the beginning line and end line. Change them into your first name and last name. Remember the strings are case-sensitive, so capitalize the first letter of the word and follow exactly the syntax of the example.

You can see lots of similar code blocks in the file. You are supposed to fill your answer between those special beginning and ending comment lines. You can insert and edit multiple lines between special comment lines in anyways, however (Important!), as the comment indicated, do not modify the special begin and comment lines **themselves**!

Let's do the second task. Scroll down to the bottom of the file and find those lines (press "shift + g" if you are using vi/vim):

```
// #8#BEGIN# DO NOT MODIFY THIS COMMENT LINE!
int banner_id = 0;
// #8#END# DO NOT MODIFY THIS COMMENT LINE!
```

Look at your student ID card, check your banner ID. Again, change **ZERO** to **your own ID**. Your unique student id will be compiled into the program, and the input of the experiment also uniquely depends on your ID.

Warning: Since every student has a unique id number, the later compiled binary file is also unique. Copy binary file from other students will be easily detected!

3. Complete Source Code (70 Points)
Read the source code and the rest comments, try to understand the function of each line of code, the basic usage of "**pthread**" library function and semaphore from the example code of producer and from the main function.

Follow the instructions in the comments and insert proper code into the rest 7 blocks to implement a producer/consumer model.

4. Run and Record Simulation Result (10 Points)
Your correct file name should be: **project5_LastName_UserID.cpp**

For example, project5_Liu_tzl0031.cpp. **Please remember to change the lastname and userID to your own ones before you continue the following steps**.

Compile your source code into a binary program. For example, use following command to include the pthread library:

$ g++ project5_Lastname_UserID.cpp -o project5_LastName_UserID -lpthread

After you compile the C++ source code successfully, please use the script command to record the running result of the program Firstname_Lastname:

$ script project5_LastName_UserID.script
Script started, file is project5_LastName_UserID.script
./project5_LastName_UserID

After you run the program, you will have the following results:

Banner id: 90.....
producer produce item 7
consumer consume item 7
producer produce item 14
consumer consume item 14
producer produce item 21
consumer consume item 21
producer produce item 28
consumer consume item 28
producer produce item 35
consumer consume item 35
producer produce item 42
consumer consume item 42

…

You should have the same result except the different in the banner ID. Then exit recording and save it into typescript file "**project5_LastName_UserID.script**" by the following command:

$ exit
exit Script done, file is project5_LastName_UserID.script

Warning: Since every student has a unique id number, the result of the simulation is also unique. Copy simulation results from other students will be easily detected!

5. Deliverables (10 Points)

Since you have generated multiple script files, please save all the script files in one directory. Make sure all the **three** files are into this folder. You should have those following files inside the folder:

1. Commands recording file: **project5_LastName_UserID.script**
2. Executable binary file: **project5_LastName_UserID**
3. Source code file: **project5_LastName_UserID.cpp**

Change the folder name to "**project5_LastName_UserID**". **Note: It is better to change the name of the folder after the above three files moved into the folder, because of the name conflict.** Please make sure the **folder name** is in a correct form. You can use the command **mv** to rename the folder:

$ mv your-current-folder-name project5_LastName_UserID

Achieve all the folder into a single tarred and compressed file with a tar command.

tar -zcvf project5_LastName_UserID.tar.gz project5_LastName_UserID

You need to submit one tarred file with a format **project5_LastName_UserID.tar.gz** through Canvas.

Grading Criteria:

1. Follow the format specification: 10%
    a. Do not break the special comments.
    b. Input your name properly (mark #0).
    c. Input your banner id properly (mark #8).
2. Complete the source code: 70%
    a. Each blank is worth 10%, in total 70% (mark #1 ~ #7).
    b. See detailed specification for each blank in the source code file.
3. Compiling and running result: 10%
    a. Compile the code successfully.
    b. Record the running results.
4. Deliverables: 10%
    a. Contains all the files.
    b. Naming all the files properly.

Late Submission Penalty:

- Late submissions will not be accepted and will result in a ZERO without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will not accept any late submission caused by internet latency.

Rebuttal Period:

You will be given **two business days** to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.