COMP 2710 – Project 5: SYNC ME NOT

Points Possible: 100
Deadline: **11:59pm, Friday, Nov 22nd, 2024 (Central Time)**

Goals:

- Able to understand coding requirements and have a logical workflow.
- Able to complete the source code to simulate producer/consumer problem.
- Able to understand the basics of the POSIX thread library.
- Able to build a binary program on a Linux machine.
- Able to run the program and record the result of the simulation.

Caution:

Read and follow the I/O format specification carefully, as the grading software will evaluate your final output based on these predefined formats. Failure to do so will result in your submission not accepted by the autograder, and you could land yourself a ZERO for this assignment.

Plagiarism is prohibited as there will be multiple ways your project can be detected if it is present (similarity of code + hash code with your banner ID). Honesty goes a long way.

SCROLL DOWN TO GET STARTED

PART A: THE PRODUCER AND CONSUMER MODEL

I. Description:

The producer and consumer model are a way to schedule how concurrent processes and threads access available resources. It contains:

- **Producer**: One or multiple processes/threads that produce data or release hardware resource.
- **Consumer**: The one process/thread that takes in data or uses hardware resource to do computations.

  Note 1: A producer could also be relatively a consumer to the output of another producer and vice versa.

- **Buffer**: The destination to store the output from producers or resources and later accessed by another consumer.

Other concepts involved in this project include:

- **POSIX thread**: Threads mechanism that satisfies POSIX standard (most operating system).
- **Mutex**: A "lock" that guaranteed that only one person has the access.

In this project, you will work with POSIX threads to simulate the producer and consumer model. `pthread` is a POSIX thread library in C++ that provides essential functions for concurrency.

For simplicity in our simulation, we assume there are only two POSIX threads: a producer and a consumer. The producer adds one unit of data (an integer) to the buffer at a time, while the consumer removes one unit of data from the buffer per operation. The buffer can hold only one unit of data at a time. The producer generates integers in a sequence (7, 14, 21, etc.) and places them in the buffer, while the consumer reads and removes these integers from the buffer.

II. Source file:

In the given source file project5_LastName_UserID.cpp, you will find the first four comment lines as follows:

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

- **TASK 1**: 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 the exact syntax of the example.

Note 2: You will 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, as the comment indicated, do not modify the special begin and comment lines themselves!

- **TASK 2**: Find these lines in the source code (`[Shift] + [G]` if you are using vi/vim) and fill in your Auburn Banner ID. Your unique student ID will be compiled into the program, and the input of the experiment also uniquely depends on your ID.

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

Note 3: Since every student has a unique Auburn Banner ID, the later compiled binary file is also unique. Copying binary file from other students will be easily detected!

- **TASK 3 - 9**: Read the source code and the rest of the comments that follow. 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 your own code into the 7 blocks to implement a producer/consumer model.

PART B: SIMULATE YOUR PROGRAM

Note 4: Remember to change the `lastname` and `userID` to your own before continuing.

Once you finish all 10 tasks in **Part A**, compile your source code into a binary program that includes the `pthread` library. Use the following command to do so:

```
$ g++ project5_King_pzk0039.cpp -o project5_King_pzk0039 -lpthread
```

Then, use the following script command to record the running result of the program:

```
$ script project5_King_pzk0039.script

Script started, file is project5_King_pzk0039.script
./project5_King_pzk0039
```

You should have the following results after running the script file, except for your banner ID:

```
Banner id: 90xxxxxxx
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
...
```

Finally, exit the recording session save it as typescript file `project5_LastName_UserID.script` by using the following command:

```
$ exit

exit Script done, file is project5_LastName_UserID.script
```

PART C: GRADING AND POLICIES

I. Grading:

100 points maximum, criteria ordered by parts, and from lowest to highest in point value.

PART A: (80 points), of which

1. (10 points) You completed Task 1 and 2.
2. (70 points) You completed Task 3 to 9.

PART B: (10 points, all carry equal point value)

1. Your code compiles successfully
2. You have a file which records the execution results

MISCELLANEOUS: (10 points, all carry equal point value)

1. Your source code is of good quality, easy to read and well-organized.
2. Use comments to provide a heading at the top of your code containing your name, Auburn Banner ID, filename, and how to compile your code. Also describe any help or sources that you used.
3. Your zipped tarball file + source code + script file + executable binary file should be named like this:

   `project5_LastName_UserID.tar.gz`

   `project5_LastName_UserID.cpp`

   `project5_LastName_UserID.script`

   `project5_LastName_UserID`

   For example, `project5_King_pzk0039.tar.gz`

   You will not lose any point if Canvas automatically changes your file name (e.g., `project5_LastName_UserID-1.tar.gz`) due to your resubmissions.

Note 5: You will automatically lose at least 40 points if there are compilation errors or warning messages when we compile your source code for both phases. You will lose points if you don't use the specific program file name, or don't have comments, or don't have a comment block on EVERY program you hand in. These will be deducted from your final score, after accounting all other requirements listed above.

II. Programming environment:

Write your program in C++. Compile and run it using AU server (no matter what kind of text editor, IDE or coding environment you use, please make sure your code could run on AU server, the only test bed we accept is the AU server).

III. Deliverables:

You must submit a tarred and compressed file – called a tarball, with an extension of `.tar.gz`, containing your `.cpp` file, script file, and executable binary file. Assuming your project is in a folder with a path of `~/comp2710/project5_King_pzk0039`, use these commands below to create a compressed file for submission:

```
cd ~/comp2710
tar vfzc project5_King_pzk0039.tar.gz project2_King_pzk0039
```

Note 6: Make sure you have all necessary files before compressing, include the followings:

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

You will lose a big chunk of points if you forget to add any of them to your submission.

IV. Late submission penalty:

- Late submissions are not accepted and will results 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.

V. Rebuttal period:

- You will be given a period of **2 business days** to read and respond to the comments and grades of your homework or project assignments. 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.

Good luck y’all! And WAR EAGLE!