Programming Assignment 3

CSC 4320/6320 - Operating Systems

Spring 2018

Dr. Xiaolin Hu

Due Date: 03/18/2018 (Sunday), 11:59pm

Problem Statement:

Read through **Project 3 (page 253-255)** in the textbook, the **Producer–Consumer Problem,** and finish it using Pthreads only (ignore the Windows API part). Your project must meet the following requirements for full credit:

1. Instead of generating random numbers and write to the buffer, producer threads will write incremental integers into the buffer. For example, if the initial value of the number is 10, then the next number should be 11, and the next will be 12....

2. Create producer and consumer threads with **3** scenarios and show your results in screenshots similar as the sample screenshots shown below. The three scenarios are: 1) more producers than consumers; 2) more consumers than producers; 3) same number of producers and consumers.

Please refer to the screenshots shown below for sample outputs under different scenarios. (Your outputs may be different)

3. The main () function will be passed **four** parameters on the command line. The four parameters are:

    1. How long will the main thread sleep before terminating
    2. The number of producer threads
    3. The number of consumer threads
    4. The initial value of the number (for producers)

Your program should be executed like this:

./buffer <*sleep time*> <*producer threads*> <*consumer threads*> <*start number*>. Here *buffer* is your executable file which is followed by four parameters, *sleep time, the number of producer threads, the number of consumer threads, and the initial number*. Eg. ./buffer 10 2 2 10

Some Notes:

1. To help you to get started, a skeleton of the buffer.c is provided in iCollege.
2. To get a thread's ID, you may call the pthread_self() function and convert the result into an integer. Below is a sample showing how a thread can get and print its ID:

    ```
    printf("Consumer %u consumed %d \n",(unsigned int)pthread_self(), random);
    ```

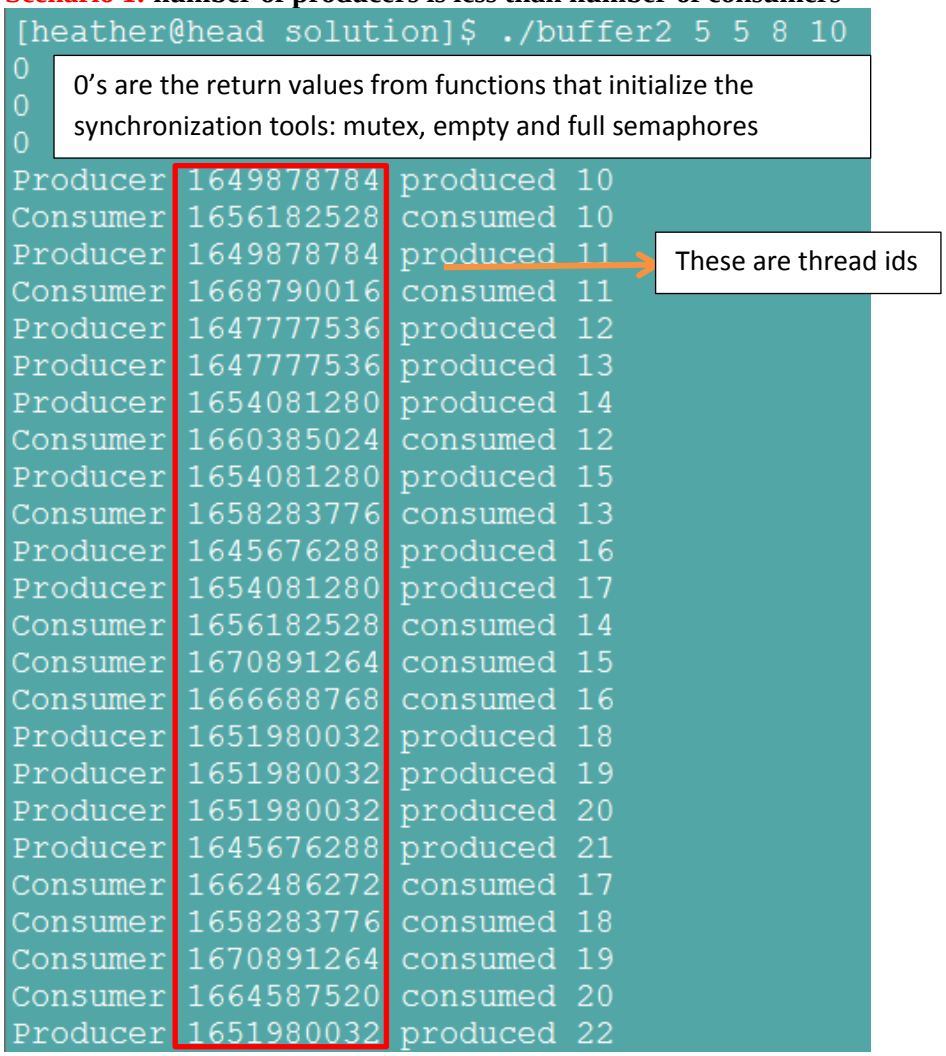
3. Using **gcc –pthread –o buffer buffer.c** to compile, and then an executable file *buffer* would be generated. (Suppose buffer.c is your code file)
4. In your report, you need to include screenshots of the three scenarios (see sample screenshots below).
5. Work individually. Programs should exhibit a modular or object-oriented design. Poor design will not earn full credit.

What to submit (submit trough iCollege)

1) Upload **a source code file** of *buffer.c* and **a project report** (named as "*HW3.pdf*" or "*HW3.doc*") to iCollege. The project report should include: 1) the source code; 2) a screenshot of the output.
2) Submit your files separately; do NOT submit a zip file.

Sample screenshots:

Scenario 1: number of producers is less than number of consumers



Scenario 2: **Number of producers is equal to number of consumers**

```
[heather@head solution]$ ./buffer2 5 8 8 15
0
0
0
Producer 1535960832 produced 15
Producer 1535960832 produced 16
Producer 1531758336 produced 17
Consumer 1552770816 consumed 15
Producer 1533859584 produced 18
Producer 1533859584 produced 19
Producer 1542264576 produced 20
Consumer 1544365824 consumed 16
Producer 1542264576 produced 21
Consumer 1552770816 consumed 17
Producer 1527555840 produced 22
Consumer 1548568320 consumed 18
Producer 1538062080 produced 23
Consumer 1546467072 consumed 19
Producer 1540163328 produced 24
Consumer 1546467072 consumed 20
Consumer 1544365824 consumed 21
Consumer 1552770816 consumed 22
Producer 1529657088 produced 25
Consumer 1550669568 consumed 23
Producer 1533859584 produced 26
Producer 1535960832 produced 27
Consumer 1556973312 consumed 24
Producer 1542264576 produced 28
Producer 1533859584 produced 29
Consumer 1546467072 consumed 25
Producer 1538062080 produced 30
Consumer 1544365824 consumed 26
Consumer 1554872064 consumed 27
Consumer 1559074560 consumed 28
Producer 1542264576 produced 31
Producer 1531758336 produced 32
```

Scenario 3: **Number of producers is larger than number of consumers**

```
[heather@head solution]$ ./buffer2 5 10 5 15
0
0
0
Producer 2146469632 produced 15
Producer 2154874624 produced 16
Producer 2156975872 produced 17
Producer 2156975872 produced 18
Producer 2142267136 produced 19
Consumer 2163279616 consumed 15
Consumer 2165380864 consumed 16
Producer 2146469632 produced 20
Producer 2146469632 produced 21
Consumer 2165380864 consumed 17
Consumer 2169583360 consumed 18
Producer 2150672128 produced 22
Consumer 2169583360 consumed 19
Consumer 2169583360 consumed 20
Producer 2154874624 produced 23
Producer 2146469632 produced 24
Producer 2152773376 produced 25
Consumer 2167482112 consumed 21
Consumer 2167482112 consumed 22
Producer 2152773376 produced 26
Producer 2150672128 produced 27
Consumer 2161178368 consumed 23
Producer 2140165888 produced 28
Consumer 2163279616 consumed 24
Consumer 2165380864 consumed 25
Producer 2144368384 produced 29
Producer 2148570880 produced 30
[heather@head solution]$
```