Georgia State University

CSc 4320/6320 Operating Systems

Fall 2020

Homework Assignment 3

Due Time: 11:59 PM, October 9

Objective:

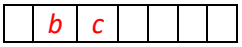
To understand and experiment with the implementation of synchronization.

Problem Statement:

Design and implement a solution for *Producer–Consumer Problem* using Pthreads library in Linux virtual machine.

Steps:

- Download *buffer.h* and the incomplete version of *hw3.c* from iCollege
- Read through Programming Project 3 of Chapter 5 (**Producer–Consumer Problem**) in the textbook.
- Follow the suggestion in the textbook to solve the problem using Pthreads. Do NOT solve the problem using Windows API.
    - Instead of implementing the buffer as a circular queue as stated in the textbook, ***implement the buffer as a Dequeue*** (Doubly Ended Queue, *i.e.* insertion and deletion can be done at both end of the queue).
    - A dequeue has different variants. In this assignment, the behaviors of consumer and producer are defined as follows:
        - The producer should add an element to the side that is farther from its queue end (right or left end); Contrarily, the consumer should remove an element from the side that is closer to its queue end (right or left end)

            For example,
            given a non-empty dequeue: 

            consumer will remove one element from the left, i.e., element *b*, because the left side is closer to its end (left end). Producer will add an element on the right, i.e., after *c*, because the right side is farther from its queue end (right end).
        - If the distances from the left-most element to the left end the distance from the right-most element to the right end are the same, it requires insertion on the right and removal on the left.

            For example,

given a non-empty dequeue: | | | *a* | *b* | *c* | | |
consumer will remove element *a*, and producer will add an element after *c*.

- If the queue is empty, the producer will add the first element to the left of the array.

- In addition to the requirement in the textbook, print a message every time an item is produced or consumed in your producer or consumer threads. The message should also include the thread ID of the producer or consumer, consumed or produced item, the insertion or removal position, and the current buffer content (see the screenshot below).
- To get a thread's ID, you may call pthread_self() function and convert the result into an integer. For example, **printf("Consumer %u",(unsigned int)pthread_self())**; will print the consumer ID.
- Compile the C source file using **gcc -pthread -o hw3 hw3.c**
- Use **./hw3 <sleep time> <# of producer threads> <# of consumer threads>** to test the program. The number of producer threads and number of consumer threads should be larger than 1. For consistency, use **./hw3 5 8 5** for the final submission. Print out the program output message after you run the program (see the screenshot below).

```
king@king-Latitude-E5540: ~/Documents/OShomework/os-ass3/dequeue
(base) king@king-Latitude-E5540:~/Documents/OShomework/os-ass3/dequeue$ ./hw3 5 8 5
Producer 2491627264 produced 8 at position 0...         ...[8, _, _, _, _]
Consumer 2541983488 consumed 8 at position 0...         ...[_, _, _, _, _]
Producer 2466449152 produced 6 at position 0...         ...[6, _, _, _, _]
Producer 2483234560 produced 9 at position 1...         ...[6, 9, _, _, _]
Producer 2474841856 produced 4 at position 2...         ...[6, 9, 4, _, _]
Producer 2500019968 produced 4 at position 3...         ...[6, 9, 4, 4, _]
Producer 2458056448 produced 8 at position 4...         ...[6, 9, 4, 4, 8]
Consumer 2533590784 consumed 6 at position 0...         ...[_, 9, 4, 4, 8]
Consumer 2525198080 consumed 8 at position 4...         ...[_, 9, 4, 4, _]
Consumer 2516805376 consumed 9 at position 1...         ...[_, _, 4, 4, _]
Consumer 2508412672 consumed 4 at position 3...         ...[_, _, 4, _, _]
Producer 2449663744 produced 7 at position 3...         ...[_, _, 4, 7, _]
Producer 2441271040 produced 1 at position 1...         ...[_, 1, 4, 7, _]
Producer 2491627264 produced 3 at position 4...         ...[_, 1, 4, 7, 3]
Consumer 2541983488 consumed 3 at position 4...         ...[_, 1, 4, 7, _]
Producer 2466449152 produced 1 at position 4...         ...[_, 1, 4, 7, 1]
Producer 2483234560 produced 5 at position 0...         ...[5, 1, 4, 7, 1]
Consumer 2525198080 consumed 5 at position 0...         ...[_, 1, 4, 7, 1]
Consumer 2533590784 consumed 1 at position 4...         ...[_, 1, 4, 7, _]
Producer 2474841856 produced 5 at position 4...         ...[_, 1, 4, 7, 5]
Producer 2500019968 produced 8 at position 0...         ...[8, 1, 4, 7, 5]
Consumer 2516805376 consumed 8 at position 0...         ...[_, 1, 4, 7, 5]
Consumer 2508412672 consumed 5 at position 4...         ...[_, 1, 4, 7, _]
Producer 2458056448 produced 9 at position 4...         ...[_, 1, 4, 7, 9]
Producer 2441271040 produced 3 at position 0...         ...[3, 1, 4, 7, 9]
Consumer 2541983488 consumed 3 at position 0...         ...[_, 1, 4, 7, 9]
Producer 2449663744 produced 9 at position 0...         ...[9, 1, 4, 7, 9]
Consumer 2525198080 consumed 9 at position 0...         ...[_, 1, 4, 7, 9]
Producer 2491627264 produced 6 at position 0...         ...[6, 1, 4, 7, 9]
Consumer 2533590784 consumed 6 at position 0...         ...[_, 1, 4, 7, 9]
Producer 2500019968 produced 3 at position 0...         ...[3, 1, 4, 7, 9]
Consumer 2516805376 consumed 3 at position 0...         ...[_, 1, 4, 7, 9]
Consumer 2508412672 consumed 9 at position 4...         ...[_, 1, 4, 7, _]
Producer 2483234560 produced 1 at position 4...         ...[_, 1, 4, 7, 1]
Producer 2474841856 produced 4 at position 0...         ...[4, 1, 4, 7, 1]
Consumer 2541983488 consumed 4 at position 0...         ...[_, 1, 4, 7, 1]
Producer 2466449152 produced 5 at position 0...         ...[5, 1, 4, 7, 1]
Consumer 2525198080 consumed 5 at position 0...         ...[_, 1, 4, 7, 1]
Producer 2441271040 produced 6 at position 0...         ...[6, 1, 4, 7, 1]
Consumer 2533590784 consumed 6 at position 0...         ...[_, 1, 4, 7, 1]
Producer 2458056448 produced 8 at position 0...         ...[8, 1, 4, 7, 1]
Consumer 2516805376 consumed 8 at position 0...         ...[_, 1, 4, 7, 1]
Producer 2449663744 produced 6 at position 0...         ...[6, 1, 4, 7, 1]
Consumer 2508412672 consumed 6 at position 0...         ...[_, 1, 4, 7, 1]
Producer 2491627264 produced 1 at position 0...         ...[1, 1, 4, 7, 1]
(base) king@king-Latitude-E5540:~/Documents/OShomework/os-ass3/dequeue$
```

Notes

- Appropriate error checking of the command line is always a good practice for programmers. It is strongly encouraged because it also reduces grading error. (You can follow the examples in the textbook and the slides).
- Design your program in a clear and modular fashion.

Submission:

Submit the following to iCollege:

- The C source file *hw3.c*
- A report in PDF or Word that includes
    - The screenshot of your program output message
    - A copy of your C source code from *hw3.c* file

Failure to follow the submission requirement will cause 10% deduction in the score.