CSC 415 - Consumer Producer Problem
Project 5

Total Points: 175 Points

Description

Project five will introduce you to POSIX synchronization primitives. You will first fix your code from homework four using the correct synchronization primitive. Now the correct output of the program should be 100.

For part two, you will need to implement the consumer-producer problem. In essence this problem is simply maintaining a shared resource that is going to be used by many threads. Some threads will add items and some will consume. It is your job to make sure each thread doesn't negatively effect the others.

Part 1 - Modifying pthread_race.c

For the first part of assignment five you will add synchronization to your solution to HW #4 to eliminate all the race conditions. You should not remove any of the nanosleep commands that you used to produce the race conditions, but simply add the necessary mutex synchronization calls for Linux threads. Your implementation should use the synchronization primitives of the programming API you are working with (Posix). Your code must be put into the pthread_race.c file inside the part1 folder. **NO OTHER FILES WILL BE GRADED**

Part 2 - Implementing Consumer Producer Problem

Part two of the assignment is (very roughly) based on Programming Project 6.40 in Silbershatz. You will be implementing a Producer-Consumer program with a bounded buffer queue of N elements, P producer threads and C consumer threads (N, P and C should be command line arguments to your program, along with three additional parameters, X, Ptime and Ctime, that are described below). Each Producer thread should Enqueue X different numbers onto the queue (sleeping for Ptime cycles in between each call to Enqueue). Each Consumer thread should Dequeue P*X/C (be careful when P*X/C is not evenly divisible) items from the queue (sleeping for Ctime cycles in between each call to Dequeue). The main program should create/initialize the Bounded Buffer Queue, print a timestamp, spawn off C consumer threads & P producer threads, wait for all of the threads to finish and then print off another timestamp & the duration of execution.

Step 1. Write high level pseudocode for the Producer and Consumer threads, as well as for the Bounded Buffer Queue (Init, Enqueue, Dequeue). Use semaphores to describe synchronization logic in your pseudocode, and put all of the calls to P/V in the Enqueue/Dequeue implementations. Submit this pseudocode in a file called pandcpseudo.txt . Design a testing strategy for verifying that all of the threads are collectively executing correctly. One possible testing strategy is to have a single atomic counter (i.e. a counter with mutex synchronization so it is guaranteed to produce different numbers) to generate numbers for Producer threads, then have the main routine combine the output from all of the Consumer threads, sort it and verify that all of the input numbers appeared as output. Submit this testing strategy as part of your design documentation.

Step 2. Implement your Producer-Consumer program using Linux threads.

For step 2, you will take your pandcpseudo.txt and implement it in C using the appropriate synchronization primitives(mutex locks and/or signal semaphores and/or count semaphores) so that your code always executes correctly. Submit well-commented source code and annotated output to demonstrate that your code is executing correctly.

The implementation has the the following requirements:

- Your implementation must be stored in pandc.c **NO OTHER .C FILES WILL BE CHECKED OR GRADED FOR ANY REASON**
- Your program must accept 6 command like arguments N,P,C,X,Ptime,Ctime Where each arguments is :
  - N is the number of buffers to maintain.
  - P is the number of producer threads.
  - C is the number of consumer threads.
  - X is the number of items each producer thread will produce.
  - Ptime is the how long each producer thread will sleep after producing an item.
  - Ctime is the how long each consumer thread will sleep after consuming an item.
- Your implementation must maintain N buffers. Size of these buffers can be one which store one integer.
- Your implementation must generate a unique sequence of numbers for the Producer threads and the Consumer threads should consume the same sequence.
- Your implementation must handle when P*X/C is not evenly divisible. For example if P = 3 and X = 5 and C = 2 then P*X/C is 15/2 which is 7.5. This means one producer thread and one consumer thread will produce and consume 1 extra item then the other thread.
- Each consumer thread must print their thread id and which item they have consumed.
- Each consumer thread must sleep for Ctime after consuming each item.

- Each producer thread must print their thread id and which item they have produced.
- Each producer thread must sleep for Ptime after producing each item.
- Your implementation must implement a test strategy that shows your program works. A simple approach is to maintain a consumer array and a producer array. If everything works out these two arrays should be identical and in order(Given enqueue and dequeue are implemented correctly).

Sample Output:

```
N = 7  P = 5 C = 3 X = 15 Ptime = 1 Ctime = 1
Current time: Sun Apr  1 20:49:12 2018

    1  was produced by producer->     1
    2  was produced by producer->     2
    3  was produced by producer->     3
    4  was produced by producer->     4
    5  was produced by producer->     5
    6  was produced by producer->     1
    1  was consumed by consumer->     1
    7  was produced by producer->     2
    2  was consumed by consumer->     1
    3  was consumed by consumer->     3
    8  was produced by producer->     3
    4  was consumed by consumer->     1
    9  was produced by producer->     4
    5  was consumed by consumer->     1
    6  was consumed by consumer->     3
    7  was consumed by consumer->     2
   10  was produced by producer->     1
    8  was consumed by consumer->     2
   11  was produced by producer->     2
   12  was produced by producer->     3
   13  was produced by producer->     4
   14  was produced by producer->     5
    9  was consumed by consumer->     1
   10  was consumed by consumer->     3
   15  was produced by producer->     1
   11  was consumed by consumer->     3
   16  was produced by producer->     2
   12  was consumed by consumer->     3
```

```
13  was consumed by consumer->     1
14  was consumed by consumer->     2
17  was produced by producer->     4
15  was consumed by consumer->     1
18  was produced by producer->     5
16  was consumed by consumer->     1
19  was produced by producer->     2
20  was produced by producer->     3
17  was consumed by consumer->     2
18  was consumed by consumer->     3
21  was produced by producer->     4
19  was consumed by consumer->     2
22  was produced by producer->     5
20  was consumed by consumer->     1
23  was produced by producer->     2
21  was consumed by consumer->     1
22  was consumed by consumer->     3
24  was produced by producer->     1
23  was consumed by consumer->     1
24  was consumed by consumer->     2
25  was produced by producer->     5
26  was produced by producer->     5
27  was produced by producer->     2
28  was produced by producer->     1
29  was produced by producer->     4
25  was consumed by consumer->     1
30  was produced by producer->     3
26  was consumed by consumer->     1
27  was consumed by consumer->     2
28  was consumed by consumer->     3
31  was produced by producer->     2
29  was consumed by consumer->     3
32  was produced by producer->     1
30  was consumed by consumer->     3
33  was produced by producer->     3
31  was consumed by consumer->     3
32  was consumed by consumer->     2
33  was consumed by consumer->     1
34  was produced by producer->     2
34  was consumed by consumer->     1
35  was produced by producer->     1
35  was consumed by consumer->     1
```

```
36  was produced by producer->     3
36  was consumed by consumer->     1
37  was produced by producer->     4
37  was consumed by consumer->     1
38  was produced by producer->     5
38  was consumed by consumer->     1
39  was produced by producer->     2
39  was consumed by consumer->     1
40  was produced by producer->     1
40  was consumed by consumer->     1
41  was produced by producer->     3
41  was consumed by consumer->     1
42  was produced by producer->     4
43  was produced by producer->     1
44  was produced by producer->     5
45  was produced by producer->     3
46  was produced by producer->     2
47  was produced by producer->     4
42  was consumed by consumer->     2
48  was produced by producer->     1
43  was consumed by consumer->     2
44  was consumed by consumer->     1
49  was produced by producer->     5
45  was consumed by consumer->     1
50  was produced by producer->     3
46  was consumed by consumer->     1
47  was consumed by consumer->     2
48  was consumed by consumer->     3
51  was produced by producer->     4
49  was consumed by consumer->     2
52  was produced by producer->     1
50  was consumed by consumer->     2
53  was produced by producer->     3
51  was consumed by consumer->     2
54  was produced by producer->     2
52  was consumed by consumer->     2
55  was produced by producer->     5
53  was consumed by consumer->     2
54  was consumed by consumer->     3
56  was produced by producer->     3
55  was consumed by consumer->     3
57  was produced by producer->     2
```

```
   56  was consumed by consumer->     3
   58  was produced by producer->     5
   57  was consumed by consumer->     3
   59  was produced by producer->     3
   58  was consumed by consumer->     3
   60  was produced by producer->     2
   59  was consumed by consumer->     3
   61  was produced by producer->     5
   62  was produced by producer->     5
   63  was produced by producer->     4
   64  was produced by producer->     3
   65  was produced by producer->     2
   66  was produced by producer->     1
   60  was consumed by consumer->     2
   61  was consumed by consumer->     3
   67  was produced by producer->     5
   62  was consumed by consumer->     2
   63  was consumed by consumer->     3
   68  was produced by producer->     4
   64  was consumed by consumer->     3
   69  was produced by producer->     1
   65  was consumed by consumer->     3
   70  was produced by producer->     3
   66  was consumed by consumer->     3
   67  was consumed by consumer->     2
   71  was produced by producer->     1
   68  was consumed by consumer->     3
   72  was produced by producer->     4
Producer Thread joined:  1
Producer Thread joined:  2
Producer Thread joined:  3
   69  was consumed by consumer->     3
   73  was produced by producer->     5
   74  was produced by producer->     4
   75  was produced by producer->     4
   70  was consumed by consumer->     2
Producer Thread joined:  4
Producer Thread joined:  5
Consumer Thread joined:  1
   71  was consumed by consumer->     2
   72  was consumed by consumer->     2
   73  was consumed by consumer->     2
   74  was consumed by consumer->     2
   75  was consumed by consumer->     2
Consumer Thread joined:  2
Consumer Thread joined:  3
Current time: Sun Apr  1 20:50:33 2018

Producer Array  | Consumer Array
1               | 1
2               | 2
3               | 3
4               | 4
5               | 5
6               | 6
7               | 7
8               | 8
9               | 9
10              | 10
11              | 11
12              | 12
13              | 13
14              | 14
15              | 15
16              | 16
17              | 17
18              | 18
19              | 19
20              | 20
21              | 21
22              | 22
23              | 23
24              | 24
25              | 25
26              | 26
27              | 27
28              | 28
29              | 29
30              | 30
31              | 31
32              | 32
33              | 33
34              | 34
35              | 35
```

```
36               | 36
37               | 37
38               | 38
39               | 39
40               | 40
41               | 41
42               | 42
43               | 43
44               | 44
45               | 45
46               | 46
47               | 47
48               | 48
49               | 49
50               | 50
51               | 51
52               | 52
53               | 53
54               | 54
55               | 55
56               | 56
57               | 57
58               | 58
59               | 59
60               | 60
61               | 61
62               | 62
63               | 63
64               | 64
65               | 65
66               | 66
67               | 67
68               | 68
69               | 69
70               | 70
71               | 71
72               | 72
73               | 73
74               | 74
75               | 75

Consume and Produce Arrays Match!
```

```
Total Runtime: 81 secs
```

What to submit

1. Updated source code for pthread_race.c for part 1.
2. pandcpseudo.txt file containing pseudo code from step 1 of Part 2.
3. Copy-n-paste program output from completed Part 2 stored in output.txt
4. Please fill in readme.txt file given for both Parts 1 and 2.
5. Push all completed code to your given repository by the deadline.

How to submit

- git add .
- git commit -m " message"
- git push

Point Breakdown

- part1 → 50 points.
- part2 step1 → 25 points.
- part2 step2 → 100 points.