RESUBMITTED:

Grade: 6/10

Please correct and resubmit.

CS 5334/4390 Spring 2017

Shirley Moore, Instructor

Homework 1

Due Tuesday, February 7

**Producer-Consumer using Pthreads Condition Variables**

Please answer the questions below about the Phtreads producer-consumer code pc.c.txt provided on the course website. You may want to modify the code to answer some of the questions. You may find it useful to refer to information in the LLNL Pthreads tutorial or Pthreads man pages.

1. Explain why the tests on fifo->full and fifo->empty are inside while loop conditions rather than if statement conditions. Describe a scenario of how the code could fail if if statements were used. You may assume multiple consumers and producers for your scenario.

For one consumer and one producer there will be no problem.

However, for multiple consumer and producer there could be a problem. The example is : Say there are 2 consumer both finds the buffer empty and starts waiting. Then producer produce something and alerts both the consumer to start work. Then first one comes and take something from buffer. So the buffer is empty now. Now again the another consumer will come and it will not check that the buffer is full or not and it will try to delete something from it. And this will create error.

2. Assume just one producer and one consumer. Would the code still be guaranteed to work correctly if just the condition variables were used and the associated mutex were not locked? If so, argue why it is still correct. If not, give an example where it fails.

No it will not work . It does not matters how many consumer or producer. If the buffer is not locked then there is a possibility that both of them tries to access the same variable at the same time .

3. Assume just one producer and one consumer. Would the code still work correctly if just one condition variable were used, rather than two – that is, both producer and consumer wait on and signal the same condition variable? If so, argue why it is still correct. If not, give an example where it fails.

Whenever producer produce something then it signals consumer and vice versa. So they operate at different time. So there should not be any problem.

4. Now assume more than one producer and/or consumer – for example, one producer and two consumers.

a. Would the code still work correctly if just one condition variable were used, rather than two. If so, argue why it is still correct. If not, give an example where it fails.

It should work because if one consumer finds it empty then it will signal the producer to be awake. Now by this time if another consumer locks the mutex and find the buffer empty it will just send the signal again but the producer is already awake. Then it will also wait and producer will produce something and signal the consumers to be awake.

Actually, the code would be incorrect. Think about when the conditional variable is signaled and who can be awakened by the signal if there is only one condition variable.

Resubmit answer:

For this case, the code will be incorrect. For this we will need at least two condition variables.

Now in the given case (one producer two consumer):

Say for example two consumers is waiting for a signal and then producer produce something and signal one of them. Then that consumer will come and make the queue empty and will signal another but the another one will come and will find it empty. So, they will be in an infinite loop.

b. As in part a, assume one condition variable is used, rather than two. Also change the code to use pthread\_cond\_broadcast() instead of pthread\_cond\_signal() in both the producer and consumer routines. Now will the code work correctly? If so, argue why it is correct. If not, give an example illustrating how it can fail. Even if it is correct, is it a good solution?

No there is a chance that one consumer finds the buffer empty and signal everyone to get awake. Then before the producer having the chance of producing some thing another consumer comes and finds the buffer still empty. Then it will signal again to everyone including the waiting consumer to be awake. Now if again the consumer take the buffer again and if the loop goes on then there will be deadlock.

Actually the code would be correct. Please try this question again.

Resubmit answer:

If pthread\_cond\_broadcast is used the code will run properly but it will be inefficient. Each time pthread\_cond\_broadcast will signal everyone (all producers and consumers), but each must wait for other (producer will wait for consumer and consumer will wait for producer) and the whole queue will not be used in this case.