

Modified Producer-Consumer Problem

In this project, you will learn about thread synchronization by implementing a variant of a producer-consumer problem. In this variant, if a producer wanting to insert into the buffer finds the buffer to be full, it does not wait (block). Instead, it leaves immediately and comes back at a later time. The consumers, however, have to wait if the buffer is empty.

Detailed Specifications:

In your program, each producer and consumer will be implemented as a separate thread. The number of producer threads, the number of consumer threads will be specified as command line parameters. The buffer is to be implemented as a FIFO circular array. The capacity of the circular array is again specified as a command line parameter.

Command Line Parameters: Your program should accept three command line parameters – Number of producer threads (henceforth represented as NP), number of consumer threads (henceforth represented as NC) and buffer capacity (BC).

System Initialization: The main thread should create a buffer with a capacity of BC. It will also create two log files – a producer log and a consumer log. Then it will create NP producer threads and NC consumer threads. Each producer is given a unique number ranging from 1 to NP. Similarly, each consumer receives a unique number ranging from 1 to NC.

Producer Thread: Each producer thread operates in an infinite loop. It produces a data item (which is essentially a string – see below for details). Then it writes an entry into the producer log file (<producer number> “Generated” <data item string>). Upon writing the log entry, it attempts to insert it into the buffer. If the insertion is successful, it puts an entry into the producer log (<producer number> <data item string> “Successful Insertion”). Then, it will sleep for a random amount of time (< 5 secs) and generate the next item. If the buffer is full, the producer will not block (wait for the buffer to have empty spaces). Instead, it leaves immediately and writes an entry into the consumer log ((<producer number> <data item string> “Buffer Full – Insertion Failed”). It will wait for random amount of time (< 2 secs), and attempts to insert the same item again. This continues until the insertion is successful after which it will continue with the sequence of operations as specified above.

Each data item is a string with the following format: <producer number>_<data item number>, where thread number is the number assigned to the thread and the <data item number> indicates the sequential number of the data item in that thread. For example, the first data

item from thread number 1 will be 1_1 and the second data item from thread number 3 will be 3_2.

Consumer Thread: Each thread also operates in an infinite loop. It first writes an entry into the consumer log file (<consumer number> “attempting to remove item from buffer”). It then attempts to remove the next item from the buffer. If successful, it will write an entry into the consumer log (<consumer number> <data item string> “successfully removed”). It will then sleep for a random amount of time (< 5 secs) and attempts to remove the next item. If the buffer is empty, the consumer thread will block until there is at least one item in the buffer at which point it will be woken up.

Buffer: The buffer needs to be implemented as a FIFO circular array. As mentioned earlier, the capacity of the array is specified as a command line parameter.

Your implementation should ensure that:

1. A produced data item should be served only to one consumer.
2. The data items are removed in FIFO order and none of the items are missed.
3. A producer is turned away only if the buffer is full.
4. A consumer waits only if the buffer is empty.
5. The program should not deadlock.

Points to note:

1. You can program in C, C++ or Java. If you are programming in C/C++ you should use the Pthreads package.
2. You are not allowed to use any thread safe data structures.
3. The writes to the log files have to be synchronized.
4. Your program should keep running until manually terminated.

What to submit?

A Zip folder containing

1. Source code
2. makefile
3. Readme file containing (1) Project partner information; (2) Any special program execution instructions.

Submissions to be done via ELC.