CS 284 Operating Systems
Programming Assignment #6

PRODUCERS/CONSUMERS (BOUNDED BUFFER) PROBLEM

** IMPORTANT NOTE: For assignments 6 and 7, You can work in groups of two **

Due: March 25th, 11:59:59 pm
==========
REQUIRED READING:
POSIX Thread Library Essentials:
http://web.umr.edu/~ercal/284/thread_handout.doc
Man Pages:
pthread_create, pthread_exit, pthread_self, pthread_join,
sched_yield, pthread_kill, tkill, pthread_cond_wait,
pthread_mutex_lock, pthread_mutex_unlock, pthread_mutex_trylock,
pthread_cond_signal, pthread_cond_init, nanosleep, etc.
==========

Disclaimer
==========

We've decided to give the class a break, and since threads are not exactly the easiest topic to fit into a shell, we won't try. For this assignment, you will be creating a brand new program (unless you really want to try to fit threads into the shell).

Be warned though, after break we will most probably be returning to our shell to add pipes and sockets.

Description
==========

This program is to demonstrate a modified version of the classical Producer/Consumer problem. This problem is as follows:

> There is a store where both Workers (Producers) and Customers (Consumers) exist. There is a cabinet in this store which all people can access. However, when one person is using the cabinet, no one else can.
>
> Workers produce objects to place within the cabinet, while Customers buy those objects, thus removing them. If there is nothing in the cabinet, then the Customer must wait until a Worker has created an object and placed that object into the cabinet. However, if the cabinet is full, then the Worker must wait until an object is bought before he/she can restock the cabinet.
>
> How can simulate this behavior?

Assignment
==========

For this assignment we will implement the Producer/Consumer problem. However, we will modify it as follows:

> A. There will be 12 Producers and 5 Consumers.
> B. There will exist 2 Cabinets (Buffers).
> C. When a widget is produced, a Producer will place that object in the

cabinet with the LEAST amount of objects in it - if possible.

D. When a Consumer is ready to buy, they will take an object from a RANDOM cabinet. If there is no objects in that cabinet, then they will try to go to the other. If there are no objects in either cabinet, then the Consumer must wait until an object becomes available.

E. Only 100 objects can be created and bought before the store can close.

So...

Write a C or C++ program that produces enough neatly formatted output statements to demonstrate that the program has correctly performed the following: (Compile and run your programs on Linux workstations in CS213 Lab)

1. Main() must:

   A. Create 12 producer threads and 5 consumer thread and then wait until they have exited.

   B. Report how many widgets each producer created and how many objects each consumer bought. The sum of the objects produced should equal the number of objects bought (each should equal 100).

   C. Produce a summary of objects created and bought. This summary should include the number of objects each producer consumed and how many objects each consumer bought.

2. The producers must:

   A. Produce a TOTAL of 100 objects then exit.

      Each object takes 5-7 seconds to create (i.e. use rand()). Please remember that rand() is NOT always MT-safe, so you may need to place a mutex around this call.

      NOTE: use nanosleep(), NOT sleep()

   B. Place a produced object in the cabinet with the LEAST number of objects. If all the cabinets are full, then the Producer must wait until a space is open. The Producer CANNOT save the finished object and begin working on another one.

   4. Declare that they have created an object and placed it into a cabinet. If the Producer is forced to wait, then state so.

3. The consumers must:

   A. Buy a TOTAL of 100 objects then exit.

      Each object takes .5 - 1.5 seconds to buy

      NOTE: Use nanosleep(), NOT usleep()

   B. Remove a bought object from a RANDOM cabinet. If there are no objects in that cabinet then the consumer will move to the other cabinet. If there are no objects in either cabinet, then the consumer must wait for an object to become available in one of the cabin

   C. Declare that they have bought an object and which object it is

```
            that was bought.  If the consumer is forced to wait, then state so.

  4.  The Cabinets (buffers) should be an array of uniquely identifed objects.
      This array is best implemented as a circular array.

      Each Cabinet can hold at most 10 objects.

      Each cabinet should be protected by its own mutex, at a minimum.  You
      may use more mutexes if you wish.

-----------------------------------
 Sound easy?  Good!

 Program design issues:
-----------------------------------
  1. Consumers should cond_wait(...) if there are no widgets available to be
     consumed
  2. Producers should cond_wait(...) if the Buffer is full.
  3. Producers and consumers should cond_signal each other to alert a waiting
     thread to wake up.
  4. No 'shared resource' should be accessed without the benefit of a
     mutex_lock().

=====================================================================

(0) Compile your programs using the "-lpthread" compiler switch:

           gcc -lpthread -o program6  program6.c

=========================================================================

You don't have to use this skeleton, but some people might find it useful.

=========================================================================

/*************************************************************************
**   This is a skeleton of the program that you need to write
**   I will try to update this if needed.
*************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h> /* POSIX threads */
#include <signal.h>

#define PRODUCER_NUM 7
#define CONSUMER_NUM 3
#define BUFFER_SIZE 8

struct SPC
{  pthread_t id;
   int num;
};

struct widget {
  pthread_t produced_by;
  int widget_no;
  int empty;
};
```

```
int prod_count = 0, cons_count = 0;
struct widget buffer[BUFFER_SIZE];
struct widget buffer2[BUFFER_SIZE];
pthread_mutex_t mutex_prod, mutex_cons, mutex_buffer, mutex_buffer2;
pthread_cond_t cond_not_full, cond_not_empty;

/*  PRODUCER */
void* producer(void* arg)
{
   int i;
   struct SPC prod;

   prod.id = ....;
   srand((long)time(0));

   while(1)
   {

     /* Check if the maximum (given) number of widgets are produced.
        If so, it is time to exit the thread.
        otherwise print a message telling that you will produce the m th widget
        IMPORTANT: Shared Variables must be examined inside a critical section
     */

     /* sleep for a while and let another thread run */

     /* Try to insert your widget into the common buffer pool.
        Use a circular buffer.

        Since buffer and the associated variables are shared
        this is a critical SECTION.

        To do this you need to use appropriate locking mechanisms.
        Use pthread_cond_wait(&cond_not_full, &mutex_buffer) if the buffer
        is FULL.

        Otherwise insert your element (a global widget number) into the next
        open buffer slot and mark it as occupied.

        Use pthread_cond_signal(&cond_not_empty) to wake up one of the
        consumers waiting for a "not_empty" buffer
     */
   }
}

/* CONSUMER */

void* consumer(void* arg)
{
   int i;
   struct SPC cons;

   cons.id = *(int*)arg;
   srand((long)time(0));

   while(1)
   {
     /* Check if the maximum (given) number of widgets are consumed.
        If so, it is time to exit this thread.
        IMPORTANT: Shared Variables must be examined inside a critical section
```

```
     */

     /* Try to consume (read/take) a widget from the circular buffer pool.
        Since buffer and the associated variables are shared
        this is a critical SECTION.

        To do this you need to use appropriate locking mechanisms.
        Use pthread_cond_wait(&cond_not_empty, &mutex_buffer) if the buffer
        is EMPTY.

        Otherwise read the buffer and mark it as empty.
        print a message telling that you consumed the k th widget

        Use pthread_cond_signal(&cond_not_full) to wake up one of the
        producers waiting for an "empty" buffer slot.
     */

   }
}

/*  MAIN  */

int main(int argc, char *argv[], char *env[])
{
   void* status;
   int prod_id[PRODUCER_NUM] = {1, 2, 3, 4, 5, 6, 7};
   int cons_id[CONSUMER_NUM] = {1, 2, 3};
   int i;

   /* ignore signal alarm */
   signal(SIGALRM, SIG_IGN);

   /* create 7 producers using pthread_create() and print a message.
      check for errors
   */

   /* create 3 consumers using pthread_create() and print a message.
      check for errors
   */

   /* wait for 7 producers and 3 consumers to finish their job */
   /* print appropriate messages as they join                  */
   /* print how many widgets are produced or consumed by each thread
    */

}


Here is a Sample Output from your program:
-----------------------------------------
**************************************************************************
*  IMPORTANT NOTE:                                                       *
*  The skeleton above and the sample output below DO NOT reflect         *
*  your complete assignment
*  Your output should contain more information about widgets being consumed *
**************************************************************************

=======================================================================

IMPORTANT NOTE:
```

```
      Try to use a nice format which is similar to the following
      sample output from last year's assignment.
      But, make sure that "SUMMARY" part is still there!
      It would also be nice to print the cabinet number where the widget is.

=====================================================================

**************************************************************************
*  IMPORTANT NOTE:                                                       *
*  The skeleton above and the sample output below DO NOT reflect         *
*  the new requirements added later in the assignment (Items 6 and 7)    *
*  Your output should contain more information about widgets being consumed *
**************************************************************************

main(): I will create 5 producers.
main(): I will create 2 consumers.
PRODUCER-1: I will produce the 1th widget.
PRODUCER-2: I will produce the 2th widget.
PRODUCER-3: I will produce the 3th widget.
PRODUCER-4: I will produce the 4th widget.
PRODUCER-5: I will produce the 5th widget.
        CONSUMER-1: I will consume the 1th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 2th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-1: I will produce the 6th widget.
PRODUCER-2: I will produce the 7th widget.
PRODUCER-3: I will produce the 8th widget.
PRODUCER-4: I will produce the 9th widget.
PRODUCER-5: I will produce the 10th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=1
        CONSUMER-1: I consumed buffer[1]: produced_by=2, widget_no=2
        CONSUMER-2: I will consume the 3th widget.
        CONSUMER-2: I consumed buffer[2]: produced_by=3, widget_no=3
        CONSUMER-1: I will consume the 4th widget.
        CONSUMER-1: I consumed buffer[3]: produced_by=4, widget_no=4
        CONSUMER-1: I will consume the 5th widget.
        CONSUMER-1: I consumed buffer[4]: produced_by=5, widget_no=5
        CONSUMER-2: I will consume the 6th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-5: I will produce the 11th widget.
PRODUCER-3: I will produce the 12th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=5, widget_no=10
        CONSUMER-2: I will consume the 7th widget.
        CONSUMER-2: I consumed buffer[1]: produced_by=3, widget_no=8
        CONSUMER-1: I will consume the 8th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 9th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 13th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=9
        CONSUMER-2: I will consume the 10th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 14th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=13
        CONSUMER-2: I will consume the 11th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 15th widget.
PRODUCER-1: I will produce the 16th widget.
PRODUCER-5: I will produce the 17th widget.
```

```
        CONSUMER-2: I consumed buffer[0]: produced_by=2, widget_no=7
        CONSUMER-1: I consumed buffer[1]: produced_by=1, widget_no=6
        CONSUMER-1: I will consume the 12th widget.
        CONSUMER-1: I consumed buffer[2]: produced_by=5, widget_no=11
        CONSUMER-2: I will consume the 13th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 14th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-5: I will produce the 18th widget.
PRODUCER-3: I will produce the 19th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=5, widget_no=17
        CONSUMER-2: I consumed buffer[1]: produced_by=3, widget_no=12
        CONSUMER-2: I will consume the 15th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 16th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 20th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=2, widget_no=15
        CONSUMER-1: I will consume the 17th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-3: I will produce the 21th widget.
PRODUCER-5: I will produce the 22th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=3, widget_no=19
        CONSUMER-2: I consumed buffer[1]: produced_by=5, widget_no=18
        CONSUMER-1: I will consume the 18th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 19th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 23th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=14
        CONSUMER-2: I will consume the 20th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-1: I will produce the 24th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=16
        CONSUMER-2: I will consume the 21th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-1: I will produce the 25th widget.
PRODUCER-4: I will produce the 26th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=1, widget_no=24
        CONSUMER-2: I consumed buffer[1]: produced_by=4, widget_no=23
        CONSUMER-1: I will consume the 22th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 23th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 27th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=2, widget_no=20
PRODUCER-5: I will produce the 28th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=5, widget_no=22
        CONSUMER-1: I will consume the 24th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 25th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 29th widget.
PRODUCER-3: I will produce the 30th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=26
        CONSUMER-1: I consumed buffer[1]: produced_by=3, widget_no=21
        CONSUMER-2: I will consume the 26th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 27th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
```

```
PRODUCER-2: I will produce the 31th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=2, widget_no=27
PRODUCER-1: I will produce the 32th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=25
        CONSUMER-1: I will consume the 28th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 29th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-3: I will produce the 33th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=3, widget_no=30
        CONSUMER-2: I will consume the 30th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-1: I will produce the 34th widget.
PRODUCER-5: I will produce the 35th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=32
        CONSUMER-1: I consumed buffer[1]: produced_by=5, widget_no=28
        CONSUMER-1: I will consume the 31th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 32th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 36th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=29
PRODUCER-1: I will produce the 37th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=1, widget_no=34
        CONSUMER-2: I will consume the 33th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 34th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 38th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=2, widget_no=31
        CONSUMER-1: I will consume the 35th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-3: I will produce the 39th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=3, widget_no=33
PRODUCER-5: I will produce the 40th widget.
PRODUCER-1: I will produce the 41th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=5, widget_no=35
        CONSUMER-1: I will consume the 36th widget.
        CONSUMER-1: I consumed buffer[1]: produced_by=1, widget_no=37
        CONSUMER-2: I will consume the 37th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 38th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
PRODUCER-4: I will produce the 42th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=4, widget_no=36
        CONSUMER-2: I will consume the 39th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-3: I will produce the 43th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=3, widget_no=39
PRODUCER-4: I will produce the 44th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=4, widget_no=42
        CONSUMER-1: I will consume the 40th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 41th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 45th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=2, widget_no=38
PRODUCER-5: I will produce the 46th widget.
        CONSUMER-1: I consumed buffer[0]: produced_by=5, widget_no=40
        CONSUMER-1: I will consume the 42th widget.
```

```
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 43th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-1: I will produce the 47th widget.
PRODUCER-4: I will produce the 48th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=41
        CONSUMER-1: I consumed buffer[1]: produced_by=4, widget_no=44
        CONSUMER-1: I will consume the 44th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-2: I will consume the 45th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-3: I will produce the 49th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=3, widget_no=43
        CONSUMER-2: I will consume the 46th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
PRODUCER-2: I will produce the 50th widget.
        CONSUMER-2: I consumed buffer[0]: produced_by=2, widget_no=45
        CONSUMER-1: I consumed buffer[0]: produced_by=3, widget_no=49
main(): PRODUCER-3 joined me. it produced 10 widgets.
        CONSUMER-2: I will consume the 47th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 48th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-1: I consumed buffer[0]: produced_by=5, widget_no=46
main(): PRODUCER-5 joined me. it produced 10 widgets.
        CONSUMER-2: I consumed buffer[0]: produced_by=1, widget_no=47
main(): PRODUCER-1 joined me. it produced 10 widgets.
        CONSUMER-2: I will consume the 49th widget.
        CONSUMER-2: The buffer is empty, I am waiting.
        CONSUMER-1: I will consume the 50th widget.
        CONSUMER-1: The buffer is empty, I am waiting.
        CONSUMER-1: I consumed buffer[0]: produced_by=4, widget_no=48
main(): PRODUCER-4 joined me. it produced 11 widgets.
main(): CONSUMER-1 joined me. it consumed 23 widgets.
        CONSUMER-2: I consumed buffer[0]: produced_by=2, widget_no=50
main(): PRODUCER-2 joined me. it produced 9 widgets.
main(): CONSUMER-2 joined me. it consumed 27 widgets.
```