Description of Producer Consumer program

At the top of the program, I include references that I used to help me make this program. After that I have standard include directives for my program and defined a constant BUFFERSIZE which will be used later on in the program. Following the program headers, I have a few global variables my program will be using. The buffer variable is the shared resource that the threads will be utilizing. I have two conditional variables, one for the producer and one for the consumer. Lastly, I have one mutex.

In main, I create 7 pthread objects t1-t7. Following the the pthreads, I create each thread manually with pthread\_create(). The arguments passed are the address of the thread, null, producer/consumer, and null. As you can see, I have 4 threads working as a producer and 3 threads working as a consumer. Following the creation of the threads I have a pthread\_exit command which waits for the threads to finish execution before ending main.

In my producer function I first initialize variables I and j. These are both used as counter variables. I have a while loop that runs while j < 3 which means the producer will produce 3 times. Inside the loop I increment j. This could have been expressed as a for loop but I just left it as a while loop. The first thing that happens in the while loop is I lock the mutex. This means the value of the mutex goes from 1 to 0 meaning that if another thread tries to use the mutex, it will be blocked until the mutex is available. The next statement is a while loop. While the buffer = 3 which is full, the producer must wait for something to produce. Then I increment the buffer and print out x’s according to the integer value in buffer. J gets incremented and a signal is sent telling any waiting consumer it is now okay to proceed. Finally, the mutex is unlocked which unblocks any waiting threads. Pthread\_exit was added at the end of the function to fix errors generated when using -Wall and -Werror to compile.

The consumer function is much like the producer function. First, I and j are initialized. A while loop runs while j < 3 which lets the consumer consume 3 times. Again, this could have been written as a for loop but I left it as a while loop. The first thing that happens is the mutex is locked which blocks other threads trying to use the mutex. Now, while the buffer = 0 which is empty, the consumer must wait for something to consume. If there is something to consume, the buffer gets decremented and print out x’s according to the value in buffer. J gets incremented and a signal is sent telling any waiting producer it is now okay to proceed. The mutex is unlocked so any waiting threads are unblocked. Pthread\_exit was added at the end of the function to fix errors generated when using -Wall and -Werror to compile.