Chinmay Ratnaparkhi
Lab Assignment 02
EECS 678 - Introduction to Operating Systems
March 8th, 2015

1. **Briefly describe the problem you are trying to solve and describe the design of your solution.**

   This lab revolves around understanding the fundamental behavior of a producer/consumer problem. The overall architecture of a producer/consumer program involves a thread generating data called the 'producer' and another thread that is using the aforementioned data, called the 'consumer'. The process needs to be synchronized since both of these thread are using a common storage, i.e. the queue where the data is stored and retrieved from. The easiest way to have the producer and consumer not access the data at the same time is to restrict their executions with lock to ensure mutual exclusion. One thread acquires the lock and until it is released, the other thread has to wait.

2. **Discuss why the busy-wait solution is inefficient for threads running on the same CPU, even though it produces the "desired behavior".**

   The spin-lock solution is not very efficient because even though a process has to wait, it is still scheduled and the the while condition on it is being evaluated constantly until the lock is released by the other process. It is certainly better to make the process sleep until the time is good for it to start working. In this manner, pthread_cond_t variable is used so that the scheduler will not line up the process but at all, saving resources. to achieve this.

3. **Why are you confident your solution is correct?**

   It is difficult to state outright that the solution is correct in case of a multithreaded program. Since the operating system controls scheduling, the output is very flexible depending upon the decisions made by the operating system rather than the programmer.

   By looking at the output generated by the program, however, it is clear that each producer and consumer operates on each unique item. Items are produced, indexed from 1 to the maximum number possible without skipping any number in between and similarly, each item produced has been consumed. Each thread is entered and exited correctly. This verification suffices the requirements to state that the solution is correct.