Summary:

In this solution I used three semaphores; mutex, fillCount, and emptyCount to solve the bounded buffer problem.

The fillCount semaphore is the number fo items that are in the buffer available for reading. The emptyCount semaphore is the number of new spaces available in the buffer where items can be written. The fillCount semaphore is incremented and the emptyCount is decremented when a new item is put into the buffer. If the producer tries to decrement emptyCount when the value is zero, the producer is put to sleep. Next time an item is consumed, emptyCount is incremented and then the producer wakes up. The consumer works the same.

Discoveries:

Threads on different operating systems (Unix versus Linux) don’t behave the same when it comes to printing at high rates. I found it much easier to look at the output of the code when using a number that increments to be placed into the buffer versus a random number.