Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

**For my producer/consumer implementation,  I utilized pthreads with a structure with shared counters and a mutex lock to control access and modification of the buffer.**

```
struct counter{
    int count;
    int in;
    int out;
    pthread_mutex_t lock;
}key;
```

**I constructed my functions so that they race for the lock (once they're out of their sleep cycle), check the count, and if it's within the correct parameters (under BUFFER_SIZE for insert, over 0 for remove), they act accordingly, and pull the appropriate items out of the buffer. The count is never read nor written outside of the lock.**

```
int insert_item(buffer_item item) {
    if(pthread_mutex_trylock(&key.lock)==0){

        if(key.count<BUFFER_SIZE){
            buffer[key.in]=item;
            int i = key.in;
            key.in=(i+1)%BUFFER_SIZE;
            int n = key.count;
            key.count = n + 1;
            pthread_mutex_unlock(&key.lock);

            return 0;
        }

        pthread_mutex_unlock(&key.lock);

        return 1;
    }
    return 1;
}
```

**The remove and insert functions are essentially identical, with the exception of their indexes.**

```
int remove_item(buffer_item *item){

    if(pthread_mutex_trylock(&key.lock)==0){

        if(key.count>0){
            *item=(buffer[key.out]);
            int n = key.count;
            key.count = n-1;
```

Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

```
            int i=key.out;
            key.out=(i+1)%BUFFER_SIZE;
            pthread_mutex_unlock(&key.lock);

            return 0;
        }

        pthread_mutex_unlock(&key.lock);

        return 1;
    }
    return 1;
}



void *producer(void *param) {
    buffer_item item;
    int p=*(int*)param;
    sleep((time_t)(rand()%s));
    while (the_murder_flag) {
        item = rand();
        if (insert_item(item)==0){
            printf("Producer #%d produced %d \n",p,item);
            sleep((time_t) s);
        }
    }
    printf("Producer #%d shutting down, good night sweet prince.\n",p);
    pthread_exit(NULL);
}
```

**Producer and consumer are basically identical as well, and are simple loops that exit based on the kill handler.**

**The remainder of my code is simply the main() setting up threads for producers and consumers, setting up the handler, and then joining.**

**Sample output:**

```
haezriel@haezriel-X200CA:~/Desktop/CS460/CS460P2$ ./procon 5 20 20
Consumer #4 consumed 1185847874          //Sometimes Consumer will print before producer,
Producer #16 produced 1185847874         //making this kind of output.
Producer #15 produced 499554076
Producer #4 produced 1474454083
```

Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

```
Consumer #5 consumed 499554076
Consumer #2 consumed 1474454083
Producer #0 produced 1398972497
Producer #6 produced 1074212492
Consumer #13 consumed 1074212492        //Here, even though the items are consumed in FIFO,
Consumer #15 consumed 1494259229        //the threads can output in random order.
Consumer #16 consumed 1392956036
Producer #11 produced 1494259229
Producer #10 produced 1392956036
Consumer #10 consumed 1398972497
Producer #14 produced 1218824924
Consumer #7 consumed 1218824924
Producer #1 produced 1598946481
Producer #3 produced 1236582642
Consumer #18 consumed 1598946481
Consumer #11 consumed 1236582642
Producer #7 produced 1430617134
Producer #9 produced 1727862074
Producer #12 produced 190258297
Producer #13 produced 772621249
Producer #8 produced 1648883559
^C
Interrupt Signal Detected, setting kill flag.
Producer #17 shutting down, good night sweet prince.
Consumer #3 shutting down, good night sweet prince.
Consumer #6 shutting down, good night sweet prince.
Consumer #1 shutting down, good night sweet prince.
Producer #19 shutting down, good night sweet prince.
Consumer #17 shutting down, good night sweet prince.
Consumer #19 shutting down, good night sweet prince.
Producer #2 shutting down, good night sweet prince.
Producer #5 shutting down, good night sweet prince.
Consumer #9 shutting down, good night sweet prince.
Consumer #8 shutting down, good night sweet prince.
Consumer #12 shutting down, good night sweet prince.
Consumer #14 shutting down, good night sweet prince.
Consumer #0 shutting down, good night sweet prince.
Producer #18 shutting down, good night sweet prince.
Consumer #4 shutting down, good night sweet prince.
Producer #16 shutting down, good night sweet prince.
Producer #15 shutting down, good night sweet prince.
Consumer #2 shutting down, good night sweet prince.
Producer #4 shutting down, good night sweet prince.
Consumer #5 shutting down, good night sweet prince.
```

Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

```
Producer #0 shutting down, good night sweet prince.
Consumer #15 shutting down, good night sweet prince.
Consumer #16 shutting down, good night sweet prince.
Producer #11 shutting down, good night sweet prince.
Producer #10 shutting down, good night sweet prince.
Consumer #10 shutting down, good night sweet prince.
Producer #14 shutting down, good night sweet prince.
Producer #6 shutting down, good night sweet prince.
Consumer #13 shutting down, good night sweet prince.
Consumer #7 shutting down, good night sweet prince.
Producer #1 shutting down, good night sweet prince.
Producer #3 shutting down, good night sweet prince.
Consumer #18 shutting down, good night sweet prince.
Consumer #11 shutting down, good night sweet prince.
Producer #7 shutting down, good night sweet prince.
Producer #9 shutting down, good night sweet prince.
Producer #12 shutting down, good night sweet prince.
Producer #13 shutting down, good night sweet prince.
Producer #8 shutting down, good night sweet prince.
All threads are closed, exiting.
```

**To demonstrate that tasks are produced and consumed in the correct order, here's another run with sequential output:**

```
haezriel@haezriel-X200CA:~/Desktop/CS460/CS460P2$ ./procon 5 20 20 d
Producer produced 639749268
Consumer consumed 639749268
Producer produced 1125908541
Consumer consumed 1125908541
Producer produced 1363135953
Producer produced 929979584
Producer produced 233132745
Producer produced 79932171
Consumer consumed 1363135953
Producer produced 760534521
Consumer consumed 929979584
Consumer consumed 233132745
Consumer consumed 79932171
Consumer consumed 760534521
Producer produced 1607580493
Consumer consumed 1607580493
Producer produced 1834169025
Consumer consumed 1834169025
Producer produced 90022233
```

Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

```
Consumer consumed 90022233
Producer produced 1863157079
Consumer consumed 1863157079
Producer produced 379294894
Consumer consumed 379294894
Producer produced 1805150139
Consumer consumed 1805150139
Producer produced 320554754
Consumer consumed 320554754
Producer produced 1525714787
Consumer consumed 1525714787
Producer produced 136686141
Consumer consumed 136686141
^C
Interrupt Signal Detected, setting kill flag.
Producer #16 shutting down, good night sweet prince.
Producer #17 shutting down, good night sweet prince.
Consumer #0 shutting down, good night sweet prince.
Consumer #3 shutting down, good night sweet prince.
Producer #8 shutting down, good night sweet prince.
Producer #9 shutting down, good night sweet prince.
Consumer #13 shutting down, good night sweet prince.
Consumer #10 shutting down, good night sweet prince.
Producer #1 shutting down, good night sweet prince.
Consumer #19 shutting down, good night sweet prince.
Producer #4 shutting down, good night sweet prince.
Consumer #4 shutting down, good night sweet prince.
Producer #2 shutting down, good night sweet prince.
Consumer #7 shutting down, good night sweet prince.
Producer #13 shutting down, good night sweet prince.
Consumer #1 shutting down, good night sweet prince.
Consumer #16 shutting down, good night sweet prince.
Consumer #12 shutting down, good night sweet prince.
Producer #0 shutting down, good night sweet prince.
Producer #12 shutting down, good night sweet prince.
Producer #3 shutting down, good night sweet prince.
Consumer #17 shutting down, good night sweet prince.
Producer #11 shutting down, good night sweet prince.
Consumer #11 shutting down, good night sweet prince.
Producer #19 shutting down, good night sweet prince.
Consumer #15 shutting down, good night sweet prince.
Consumer #18 shutting down, good night sweet prince.
Producer #6 shutting down, good night sweet prince.
Consumer #5 shutting down, good night sweet prince.
```

Michael Allen-Bond
CS460
02/19/2016
P2 Write-Up

```
Producer #18 shutting down, good night sweet prince.
Producer #14 shutting down, good night sweet prince.
Producer #5 shutting down, good night sweet prince.
Consumer #9 shutting down, good night sweet prince.
Consumer #6 shutting down, good night sweet prince.
Producer #7 shutting down, good night sweet prince.
Consumer #2 shutting down, good night sweet prince.
Producer #15 shutting down, good night sweet prince.
Producer #10 shutting down, good night sweet prince.
Consumer #8 shutting down, good night sweet prince.
Consumer #14 shutting down, good night sweet prince.
All threads are closed, exiting.
```