

CS 537: Intro to Operating Systems (Summer 2017)

Worksheet 8 - Producers and Consumers

July 20th, 2017 (Thursday)

Here is some code for the producer/consumer problem we were trying to solve in class today. We'll use a new primitive: `Pthread_cond_broadcast()`. Unlike traditional signaling, this wakes up ALL threads waiting on a condition. Here is some code using such a broadcast:

```
void *producer(void *arg) {
    int i;
    for (i = 0; i < loops; i++) {
        Pthread_mutex_lock(&mutex);
        while (count == MAX)
            Pthread_cond_wait(&cv, &mutex);
        put(i);
        Pthread_cond_broadcast(&cv); // here!
        Pthread_mutex_unlock(&mutex);
    }
}

void *consumer(void *arg) {
    int i;
    for (i = 0; i < loops; i++) {
        Pthread_mutex_lock(&mutex);
        while (count == 0)
            Pthread_cond_wait(&cv, &mutex);
        int tmp = get();
        Pthread_cond_broadcast(&cv); // here!
        Pthread_mutex_unlock(&mutex);
        printf("%d\n", tmp);
    }
}
```

Does this code work? If **yes**, then explain **why** does this code work? If **no**, then explain **why not**?