OS ASSIGNMENT 4

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define BUFFER_SIZE 5

int buffer[BUFFER_SIZE];
sem_t empty, full;
pthread_mutex_t mutex;
int in = 0, out = 0;

void print_buffer() {
    printf("Buffer: [");
    for (int i = 0; i < BUFFER_SIZE; i++) {
        printf("%d", buffer[i]);
        if (i < BUFFER_SIZE - 1) {
            printf(", ");
        }
    }
    printf("]\n");
}

void *producer(void *arg) {
    int item;

    while(1){
        item = rand() % 100;

        sem_wait(&empty);
        pthread_mutex_lock(&mutex);

        buffer[in] = item;
        printf("Prodcer %d Produced: %d\n",in, item);
        in = (in + 1) % BUFFER_SIZE;

        pthread_mutex_unlock(&mutex);
        sem_post(&full);

        print_buffer();

        sleep(2);
    }
    pthread_exit(NULL);
}

void *consumer(void *arg) {
```

```
    int item;
    while(1) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);

        item = buffer[out];
        printf("Consumer %d Consumed: %d\n",out,item);
        out = (out + 1) % BUFFER_SIZE;


        pthread_mutex_unlock(&mutex);
        sem_post(&empty);

        print_buffer();

        sleep(5);
    }
    pthread_exit(NULL);
}

int main() {
    pthread_t producer_thread, consumer_thread;

    sem_init(&empty, 0, BUFFER_SIZE);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);

    pthread_create(&producer_thread, NULL, producer, NULL);
    pthread_create(&consumer_thread, NULL, consumer, NULL);

    pthread_join(producer_thread, NULL);
    pthread_join(consumer_thread, NULL);

    sem_destroy(&empty);
    sem_destroy(&full);
    pthread_mutex_destroy(&mutex);

    return 0;
}
```