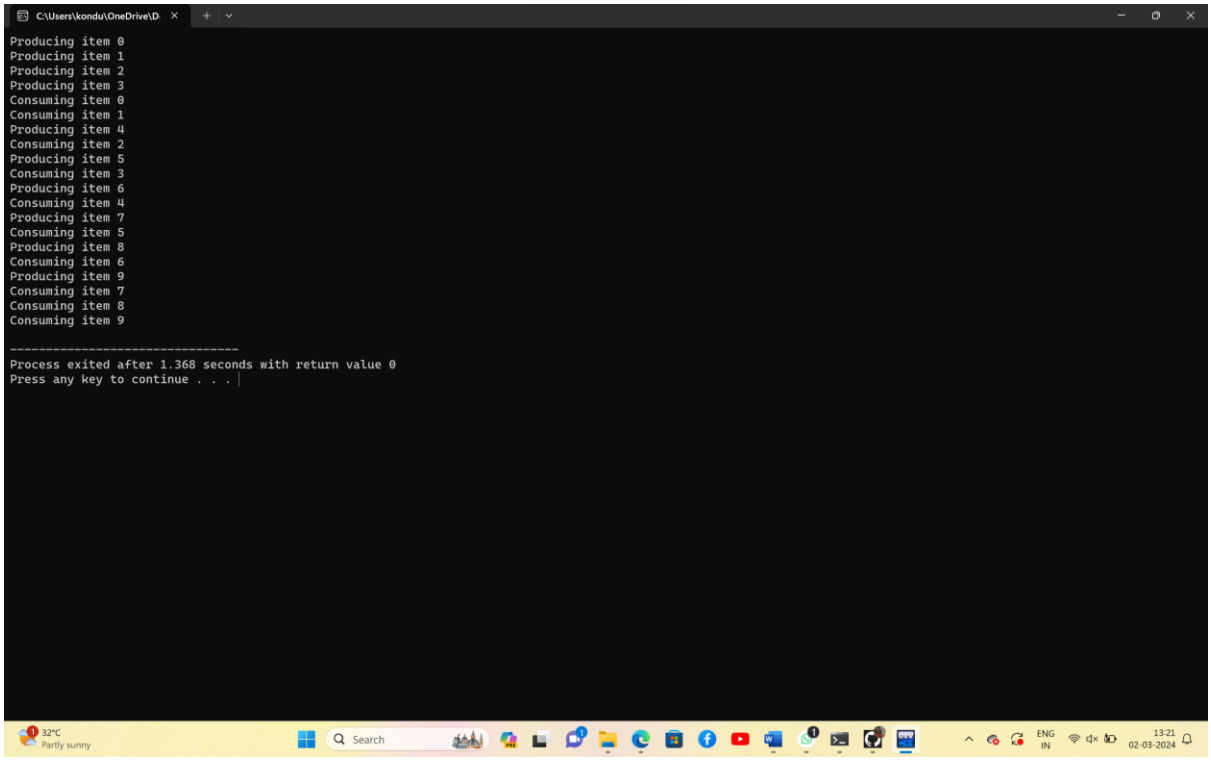
```
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>

#define BUFFER_SIZE 5
#define NUM_ITEMS 10

int buffer[BUFFER_SIZE];
int in = 0, out = 0;

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t can_produce = PTHREAD_COND_INITIALIZER;
pthread_cond_t can_consume = PTHREAD_COND_INITIALIZER;

void *producer(void *arg) {
   for (int i = 0; i < NUM_ITEMS; i++) {
      pthread_mutex_lock(&mutex);

      while (((in + 1) % BUFFER_SIZE) == out) {
         pthread_cond_wait(&can_produce, &mutex);
      }

      buffer[in] = i;
      printf("Producing item %d\n", i);
      in = (in + 1) % BUFFER_SIZE;

      pthread_cond_signal(&can_consume);
      pthread_mutex_unlock(&mutex);
   }
```

```
    return NULL;
}

void *consumer(void *arg) {
    for (int i = 0; i < NUM_ITEMS; i++) {
        pthread_mutex_lock(&mutex);

        while (out == in) {
            pthread_cond_wait(&can_consume, &mutex);
        }

        int item = buffer[out];
        printf("Consuming item %d\n", item);
        out = (out + 1) % BUFFER_SIZE;

        pthread_cond_signal(&can_produce);
        pthread_mutex_unlock(&mutex);
    }

    return NULL;
}

int main() {
    pthread_t producer_thread, consumer_thread;

    pthread_create(&producer_thread, NULL, producer, NULL);
    sleep(1); // To ensure the producer starts first
    pthread_create(&consumer_thread, NULL, consumer, NULL);

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

```
    return 0;
}
```


C:\Users\kondu\OneDrive\D
Producing item 0
Producing item 1
Producing item 2
Producing item 3
Consuming item 0
Consuming item 1
Producing item 4
Consuming item 2
Producing item 5
Consuming item 3
Producing item 6
Consuming item 4
Producing item 7
Consuming item 5
Producing item 8
Consuming item 6
Producing item 9
Consuming item 7
Consuming item 8
Consuming item 9

Process exited after 1.368 seconds with return value 0
Press any key to continue . . .
32°C
Partly sunny
Search
ENG
IN
13:21
02-03-2024