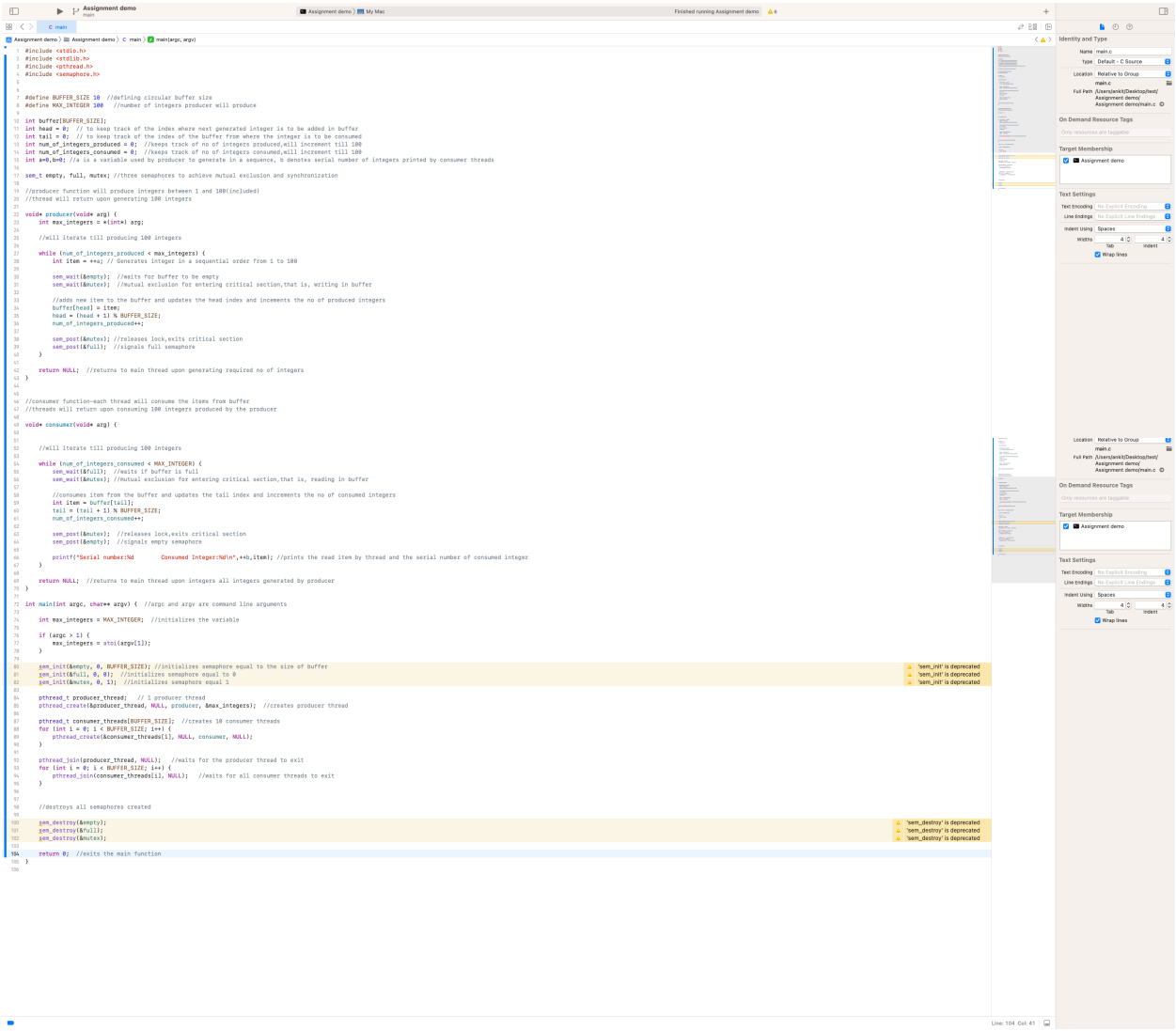
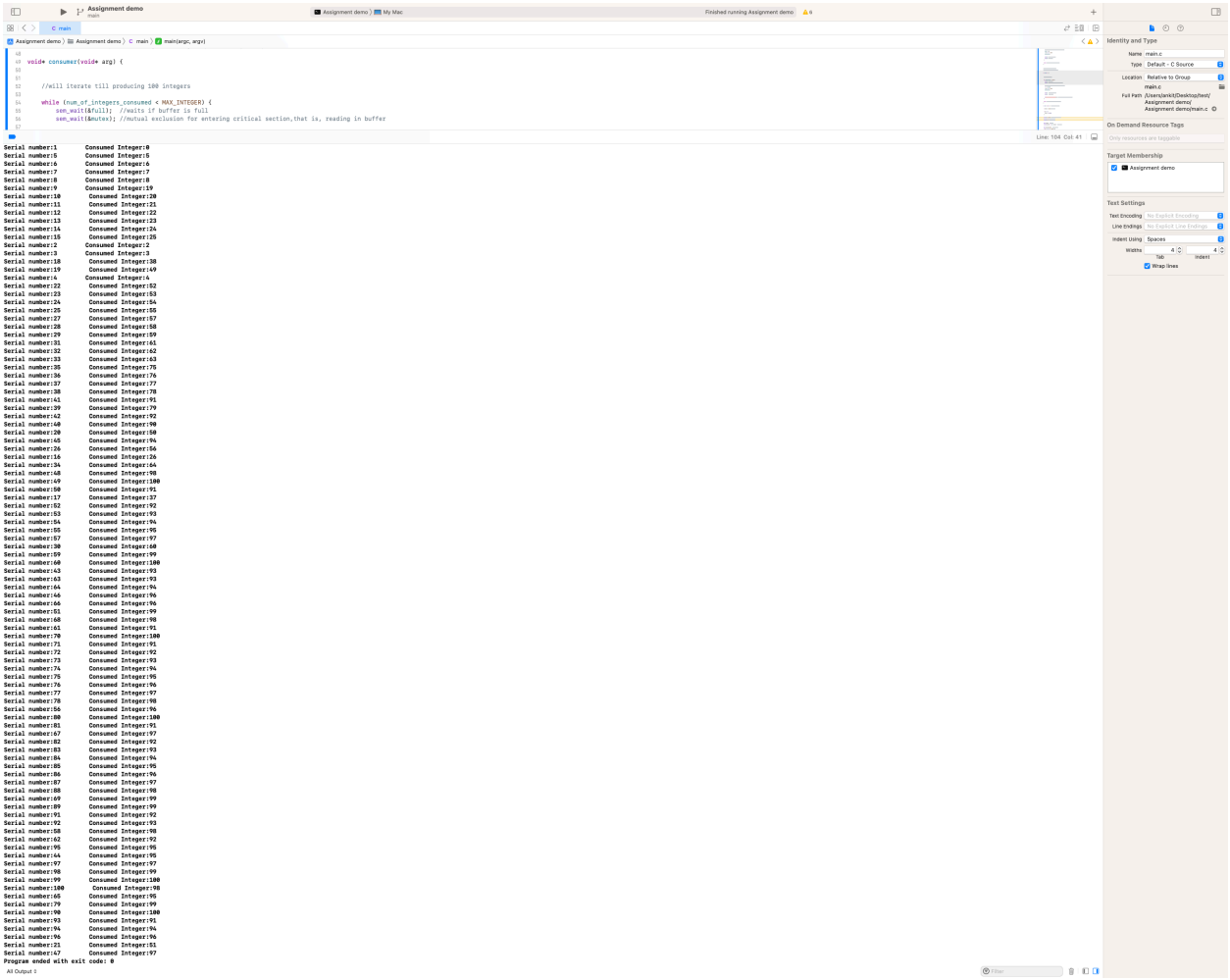


```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>


#define BUFFER_SIZE 10  //defining circular buffer size
#define MAX_INTEGER 100  //number of integers producer will produce

int buffer[BUFFER_SIZE];
int head = 0;  // to keep track of the index where next generated integer is to be added in buffer
int tail = 0;  // to keep track of the index of the buffer from where the integer is to be consumed
int num_of_integers_produced = 0;  //keeps track of no of integers produced,will increment till 100
int num_of_integers_consumed = 0;  //keeps track of no of integers consumed,will increment till 100
int a=0,b=0; //a is a variable used by producer to generate in a sequence, b denotes serial number of integers printed by consumer threads

sem_t empty, full, mutex; //three semaphores to achieve mutual exclusion and synchronization

//producer function will produce integers between 1 and 100(included)
//thread will return upon generating 100 integers

void* producer(void* arg) {
    int max_integers = *(int*) arg;

    //will iterate till producing 100 integers

    while (num_of_integers_produced < max_integers) {
        int item = ++a; // Generates integer in a sequential order from 1 to 100

        sem_wait(&empty);  //waits for buffer to be empty
        sem_wait(&mutex);  //mutual exclusion for entering critical section,that is, writing in buffer

        //adds new item to the buffer and updates the head index and incements the no of produced integers
        buffer[head] = item;
        head = (head + 1) % BUFFER_SIZE;
        num_of_integers_produced++;

        sem_post(&mutex); //releases lock,exits critical section
        sem_post(&full);  //signals full semaphore
    }

    return NULL;  //returns to main thread upon generating required no of integers
}


//consumer function-each thread will consume the items from buffer
//threads will return upon consuming 100 integers produced by the producer

void* consumer(void* arg) {


    //will iterate till producing 100 integers

    while (num_of_integers_consumed < MAX_INTEGER) {
        sem_wait(&full);  //waits if buffer is full
        sem_wait(&mutex); //mutual exclusion for entering critical section,that is, reading in buffer

        //consumes item from the buffer and updates the tail index and increments the no of consumed integers
        int item = buffer[tail];
        tail = (tail + 1) % BUFFER_SIZE;
        num_of_integers_consumed++;

        sem_post(&mutex);  //releases lock,exits critical section
        sem_post(&empty);  //signals empty semaphore

        printf("Serial number:%d        Consumed Integer:%d\n",++b,item); //prints the read item by thread and the serial number of consumed integer
    }

    return NULL;  //returns to main thread upon integers all integers generated by producer
}

int main(int argc, char** argv) {  //argc and argv are command line arguments

    int max_integers = MAX_INTEGER;  //initializes the variable

    if (argc > 1) {
        max_integers = atoi(argv[1]);
    }

    sem_init(&empty, 0, BUFFER_SIZE); //initializes semaphore equal to the size of buffer
    sem_init(&full, 0, 0);  //initializes semaphore equal to 0
    sem_init(&mutex, 0, 1);  //initializes semaphore equal 1

    pthread_t producer_thread;   // 1 producer thread
    pthread_create(&producer_thread, NULL, producer, &max_integers);  //creates producer thread

    pthread_t consumer_threads[BUFFER_SIZE];  //creates 10 consumer threads
    for (int i = 0; i < BUFFER_SIZE; i++) {
        pthread_create(&consumer_threads[i], NULL, consumer, NULL);
    }

    pthread_join(producer_thread, NULL);   //waits for the producer thread to exit
    for (int i = 0; i < BUFFER_SIZE; i++) {
        pthread_join(consumer_threads[i], NULL);   //waits for all consumer threads to exit
    }


    //destroys all semaphores created

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

    return 0;  //exits the main function
}
```

Source code


Assignment demo
main
Assignment demo 〉 My Mac
Finished running Assignment demo
4
C main
Assignment demo 〉 Assignment demo 〉 C main 〉 main(argc, argv)
void* consumer(void* arg) {
//will iterate till producing 100 integers
while (num_of_integers_consumed < MAX_INTEGER) {
sem_wait(&full); //waits if buffer is full
sem_wait(&mutex); //mutual exclusion for entering critical section,that is, reading in buffer
Line: 104 Col: 41
Serial number:1 Consumed Integer:0
Serial number:5 Consumed Integer:5
Serial number:6 Consumed Integer:6
Serial number:7 Consumed Integer:7
Serial number:8 Consumed Integer:8
Serial number:9 Consumed Integer:19
Serial number:10 Consumed Integer:20
Serial number:11 Consumed Integer:21
Serial number:12 Consumed Integer:22
Serial number:13 Consumed Integer:23
Serial number:14 Consumed Integer:24
Serial number:15 Consumed Integer:25
Serial number:2 Consumed Integer:2
Serial number:3 Consumed Integer:3
Serial number:18 Consumed Integer:38
Serial number:19 Consumed Integer:49
Serial number:4 Consumed Integer:4
Serial number:22 Consumed Integer:52
Serial number:23 Consumed Integer:53
Serial number:24 Consumed Integer:54
Serial number:25 Consumed Integer:55
Serial number:27 Consumed Integer:57
Serial number:28 Consumed Integer:58
Serial number:29 Consumed Integer:59
Serial number:31 Consumed Integer:61
Serial number:32 Consumed Integer:62
Serial number:33 Consumed Integer:63
Serial number:35 Consumed Integer:75
Serial number:36 Consumed Integer:76
Serial number:37 Consumed Integer:77
Serial number:38 Consumed Integer:78
Serial number:41 Consumed Integer:91
Serial number:39 Consumed Integer:79
Serial number:42 Consumed Integer:92
Serial number:40 Consumed Integer:90
Serial number:20 Consumed Integer:50
Serial number:46 Consumed Integer:94
Serial number:26 Consumed Integer:56
Serial number:16 Consumed Integer:26
Serial number:34 Consumed Integer:64
Serial number:48 Consumed Integer:98
Serial number:49 Consumed Integer:100
Serial number:50 Consumed Integer:91
Serial number:17 Consumed Integer:37
Serial number:52 Consumed Integer:92
Serial number:53 Consumed Integer:93
Serial number:54 Consumed Integer:94
Serial number:55 Consumed Integer:95
Serial number:57 Consumed Integer:97
Serial number:30 Consumed Integer:60
Serial number:59 Consumed Integer:99
Serial number:60 Consumed Integer:100
Serial number:43 Consumed Integer:93
Serial number:63 Consumed Integer:93
Serial number:64 Consumed Integer:94
Serial number:46 Consumed Integer:96
Serial number:66 Consumed Integer:96
Serial number:51 Consumed Integer:99
Serial number:68 Consumed Integer:98
Serial number:61 Consumed Integer:91
Serial number:70 Consumed Integer:100
Serial number:71 Consumed Integer:91
Serial number:72 Consumed Integer:92
Serial number:73 Consumed Integer:93
Serial number:74 Consumed Integer:94
Serial number:75 Consumed Integer:95
Serial number:76 Consumed Integer:96
Serial number:77 Consumed Integer:97
Serial number:78 Consumed Integer:98
Serial number:56 Consumed Integer:96
Serial number:80 Consumed Integer:100
Serial number:81 Consumed Integer:91
Serial number:67 Consumed Integer:97
Serial number:82 Consumed Integer:92
Serial number:83 Consumed Integer:93
Serial number:84 Consumed Integer:94
Serial number:85 Consumed Integer:95
Serial number:86 Consumed Integer:96
Serial number:87 Consumed Integer:97
Serial number:88 Consumed Integer:98
Serial number:69 Consumed Integer:99
Serial number:89 Consumed Integer:99
Serial number:91 Consumed Integer:92
Serial number:92 Consumed Integer:93
Serial number:58 Consumed Integer:98
Serial number:62 Consumed Integer:92
Serial number:95 Consumed Integer:95
Serial number:44 Consumed Integer:95
Serial number:97 Consumed Integer:97
Serial number:98 Consumed Integer:99
Serial number:99 Consumed Integer:100
Serial number:100 Consumed Integer:98
Serial number:65 Consumed Integer:95
Serial number:79 Consumed Integer:99
Serial number:90 Consumed Integer:100
Serial number:93 Consumed Integer:91
Serial number:94 Consumed Integer:94
Serial number:96 Consumed Integer:96
Serial number:21 Consumed Integer:51
Serial number:47 Consumed Integer:97
Program ended with exit code: 0
All Output
Filter
Identity and Type
Name main.c
Type Default - C Source
Location Relative to Group
main.c
Full Path /Users/ankitd/Desktop/test/Assignment demo/Assignment demo/main.c
On Demand Resource Tags
Only resources are taggable
Target Membership
Assignment demo
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines