Roll no -33242

OS Assignment 4 A : Producer - Consumer Problem

Code for producer - consumer problem:

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

int *buffer; // Dynamic buffer
int buffer_size;
int in = 0, out = 0;

sem_t empty;
sem_t full;
pthread_mutex_t mutex;

void print_buffer() {
    printf("Buffer: [ ");
    for (int i = 0; i < buffer_size; i++) {
        if (i == in && i == out) {
            printf("I&O ");
        } else if (i == in) {
            printf("I ");
        } else if (i == out) {
            printf("O ");
        } else if (i < in && i >= out) {
            printf("%d ", buffer[i]);
        } else if (in < out && (i >= out || i < in)) {
            printf("%d ", buffer[i]);
        } else {
            printf("E ");
        }
    }
    printf("]\n");
}

void *producer(void *arg) {
    int item_count = *((int *)arg);
    int item;
    for (int i = 0; i < item_count; i++) {
        item = rand() % 100; // Produce a random item
        sem_wait(&empty); // Wait for an empty slot
        pthread_mutex_lock(&mutex); // Lock the buffer

        // Add item to the buffer
```

```
        buffer[in] = item;
        in = (in + 1) % buffer_size;
        printf("Producer produced: %d\n", item);
        print_buffer();

        pthread_mutex_unlock(&mutex); // Unlock the buffer
        sem_post(&full); // Signal that there is a full slot

        sleep(1); // Simulate time taken to produce an item
    }
    pthread_exit(NULL);
}

void *consumer(void *arg) {
    int item_count = *((int *)arg);
    int item;
    for (int i = 0; i < item_count; i++) {
        sem_wait(&full); // Wait for a full slot
        pthread_mutex_lock(&mutex); // Lock the buffer

        // Remove item from the buffer
        item = buffer[out];
        buffer[out] = 0; // Optional: Clear the consumed slot
        out = (out + 1) % buffer_size;
        printf("Consumer consumed: %d\n", item);
        print_buffer();

        pthread_mutex_unlock(&mutex); // Unlock the buffer
        sem_post(&empty); // Signal that there is an empty slot

        sleep(1); // Simulate time taken to consume an item
    }
    pthread_exit(NULL);
}

int main() {
    pthread_t prod_thread, cons_thread;
    int produce_count, consume_count;

    // Get user input for buffer size, produce count, and consume count
    printf("Enter the buffer size: ");
    scanf("%d", &buffer_size);
    printf("Enter the number of items to produce: ");
    scanf("%d", &produce_count);
    printf("Enter the number of items to consume: ");
    scanf("%d", &consume_count);

    // Allocate buffer dynamically
    buffer = (int *)malloc(buffer_size * sizeof(int));

    // Initialize the buffer with empty slots
```

```
    for (int i = 0; i < buffer_size; i++) {
        buffer[i] = 0;
    }

    // Initialize the semaphores and mutex
    sem_init(&empty, 0, buffer_size);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);

    // Create producer and consumer threads
    pthread_create(&prod_thread, NULL, producer, &produce_count);
    pthread_create(&cons_thread, NULL, consumer, &consume_count);

    // Wait for threads to finish
    pthread_join(prod_thread, NULL);
    pthread_join(cons_thread, NULL);

    // Destroy the semaphores and mutex
    sem_destroy(&empty);
    sem_destroy(&full);
    pthread_mutex_destroy(&mutex);

    // Free the dynamically allocated buffer
    free(buffer);

    return 0;
}
```

OUTPUT-


monika@monika-VirtualBox: ~/33242
Producer produced: 92
Consumer consumed: 92
Producer produced: 49
Consumer consumed: 49
Producer produced: 21
Consumer consumed: 21
monika@monika-VirtualBox:~/33242$ gcc producer_consumer.c
monika@monika-VirtualBox:~/33242$ ./a.out
Enter the buffer size: 5
Enter the number of items to produce: 7
Enter the number of items to consume: 7
Producer produced: 83
Buffer: [O I E E E]
Consumer consumed: 83
Buffer: [E I&O E E E]
Producer produced: 86
Buffer: [E O I E E]
Consumer consumed: 86
Buffer: [E E I&O E E]
Producer produced: 77
Buffer: [E E O I E]
Consumer consumed: 77
Buffer: [E E E I&O E]
Producer produced: 15
Buffer: [E E E O I]
Consumer consumed: 15
Buffer: [E E E E I&O]
Producer produced: 93
Buffer: [I E E E O]
Consumer consumed: 93
Buffer: [I&O E E E E]
Producer produced: 35
Buffer: [O I E E E]
Consumer consumed: 35
Buffer: [E I&O E E E]
Producer produced: 86
Buffer: [E O I E E]
Consumer consumed: 86
Buffer: [E E I&O E E]
monika@monika-VirtualBox:~/33242$