Ex. No: 8
Date: 22/2/25

PRODUCER CONSUMER USING SEMAPHORES

AIM:
To write a program to implement solutions to producer consumer problem using semaphores.

ALGORITHM:

1. Initialize semaphore empty, full and mutex.
2. Create two threads- the producer thread and the consumer thread.
3. Wait for target thread termination.
4. Call sem_wait on empty semaphore followed by mutex semaphore before entry into critical section.
5. Produce/Consume the item in the critical section.
6. Call sem_post on mutex semaphore followed by full semaphore
7. before exiting the critical section.
8. Allow the other thread to enter its critical section.
9. Terminate after looping ten times in producer and consumer Threads each.

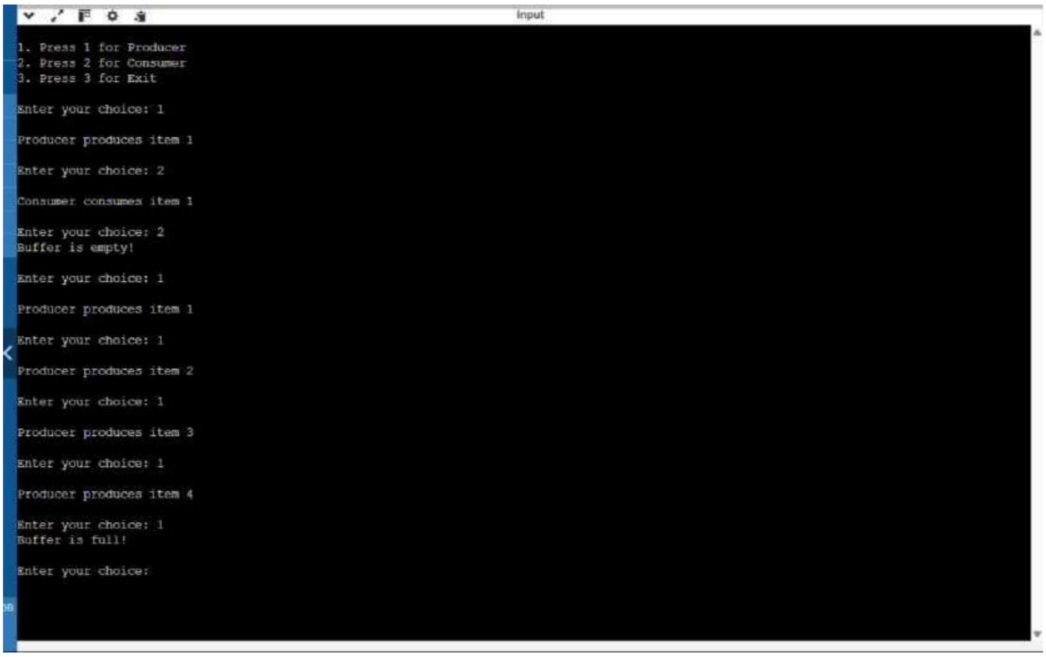
PROGRAM:

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

int mutex = 1;
int full = 0;
int empty = 10, x = 0;

pthread_mutex_t lock;

void *producer(void *arg)
{
    pthread_mutex_lock(&lock);

    if (empty != 0) {
        --mutex;
        ++full;
        --empty;
        x++;
        printf("\nProducer produces item %d\n", x);
        ++mutex;
    } else {
        printf("Buffer is full!\n");
    }

    pthread_mutex_unlock(&lock);
```

```
return NULL;
```

```
}

void *consumer(void *arg)
{
    pthread_mutex_lock(&lock);

    if (full != 0) {
        --mutex;
        --full;
        ++empty;
        printf("\nConsumer consumes item %d\n",
        x); x--;
        ++mutex;
    } else {
        printf("Buffer is empty!\n");
    }

    pthread_mutex_unlock(&lock);
    return NULL;
}

int main()
{
    int n, i;
    pthread_t prod_thread, cons_thread;
    pthread_mutex_init(&lock, NULL);
    printf("\n1. Press 1 for Producer"
           "\n2. Press 2 for Consumer"
           "\n3. Press 3 for Exit\n");

    for (i = 1; i > 0; i++) {
        printf("\nEnter your choice: ");
        scanf("%d", &n);

        switch (n) {
        case 1:
            if (mutex == 1 && empty != 0) {
                pthread_create(&prod_thread, NULL, producer, NULL);
                pthread_join(prod_thread, NULL);
            } else {
                printf("Buffer is full!\n");
            }
            break;

        case 2:
            if (mutex == 1 && full != 0) {
                pthread_create(&cons_thread, NULL, consumer, NULL);
                pthread_join(cons_thread, NULL);
            } else {
                printf("Buffer is empty!\n");
            }
            break;

        case 3:
            pthread_mutex_destroy(&lock);
            exit(0);
            break;
        default:
            printf("Invalid choice! Please enter a valid option.\n");
        }
    }

    return 0;
}
```

OUTPUT:



```
1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit

Enter your choice: 1

Producer produces item 1

Enter your choice: 2

Consumer consumes item 1

Enter your choice: 2
Buffer is empty!

Enter your choice: 1

Producer produces item 1

Enter your choice: 1

Producer produces item 2

Enter your choice: 1

Producer produces item 3

Enter your choice: 1

Producer produces item 4

Enter your choice: 1
Buffer is full!

Enter your choice:
```

RESULT:

Thus, the Producer Consumer Program using Semaphore is Successfully Implemented.