OPERATING SYSTEM - CS23431

EXP 8

PRODUCER CONSUMER PROBLEM USING SEMAPHORES

NAME: Dhanvin PR

ROLL NO: 230701071

PROGRAM:

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

#define SIZE 10

int buffer[SIZE],count=0;
sem_t empty, full, mutex;

void* producer(void* arg) {
    for (int i = 0; i < 10; i++) {
        int val;
        sem_getvalue(&empty, &val);
        if (val == 0)
            printf("Buffer is full. Producer waiting...\n");
        sem_wait(&empty);
        sem_wait(&mutex);

        if(count<SIZE)
        {
        buffer[count++]=i;
        printf("Producer produces item: %d\n", i);
        }
        sem_post(&mutex);
        sem_post(&full);

        sleep(1);
    }
    return NULL;
}
```

```
void* consumer(void* arg) {
    for (int i = 0; i < 10; i++) {
        int val;
        sem_getvalue(&full, &val);
        if (val == 0)
            printf("Buffer is empty. Consumer waiting...\n");

        sem_wait(&full);
        sem_wait(&mutex);

        if(count>0){
        printf("Consumer consumes item: %d\n", buffer[--count]);
        }
        sem_post(&mutex);
        sem_post(&empty);

        sleep(1);
    }
    return NULL;
}

int main() {
    pthread_t p, c;
    int choice;
    sem_init(&empty, 0, SIZE);
    sem_init(&full, 0, 0);
    sem_init(&mutex, 0, 1);
    while(1)
    {
    printf("1. Producer\n");
    printf("2. Consumer\n");
    printf("3. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            pthread_create(&p, NULL, producer, NULL);
            pthread_join(p, NULL);
```

```
            break;

        case 2:
            pthread_create(&c, NULL, consumer, NULL);
            pthread_join(c, NULL);
            break;
        case 3:
            printf("Exiting...");
            exit(0);
        default:
            printf("Invalid choice!! Please try again.");
            break;

    }
    }

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

    return 0;
}
```

OUTPUT: