

**Ex. No.: 8**

## **PRODUCER CONSUMER USING SEMAPHORES**

**Aim:** To write a program to implement solution to producer consumer problem using semaphores.

### **Program Code:**

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

#define SIZE 3

int buffer[SIZE];
int in = 0, out = 0, count = 0;

sem_t empty, full, mutex;

void* producer(void* arg) {
    if (sem_trywait(&empty) == 0) {
        sem_wait(&mutex);
        buffer[in] = ++count;
        printf("Producer produces the item %d\n", buffer[in]);
        in = (in + 1) % SIZE;
        sem_post(&mutex);
        sem_post(&full);
    } else {
        printf("Buffer is full!!\n");
    }
    return NULL;
}

void* consumer(void* arg) {
    if (sem_trywait(&full) == 0) {
        sem_wait(&mutex);
        int item = buffer[out];
        printf("Consumer consumes item %d\n", item);
        out = (out + 1) % SIZE;
        sem_post(&mutex);
        sem_post(&empty);
    } else {
        printf("Buffer is empty!!\n");
    }
}
```

```

    }
    return NULL;
}

int main() {
    pthread_t tid;
    int choice;

    sem_init(&empty, 0, SIZE);
    sem_init(&full, 0, 0);
    sem_init(&mutex, 0, 1);

    while (1) {
        printf("\n1. Producer\n2. Consumer\n3. Exit\nEnter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            pthread_create(&tid, NULL, producer, NULL);
            pthread_join(tid, NULL);
        } else if (choice == 2) {
            pthread_create(&tid, NULL, consumer, NULL);
            pthread_join(tid, NULL);
        } else if (choice == 3) {
            break;
        } else {
            printf("Invalid choice!\n");
        }
    }

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

    return 0;
}

```

**OUTPUT :**

```
Enter your choice: 1
Producer produces the item 1
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 2
Consumer consumes item 1
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 2
Buffer is empty!!
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 1
Producer produces the item 2
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 1
Producer produces the item 3
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 1
Producer produces the item 4
```

- 1. Producer
- 2. Consumer
- 3. Exit

```
Enter your choice: 1
Buffer is full!!
```