

Ex. No.: 8

PRODUCER CONSUMER USING SEMAPHORES

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

Algorithm:

1. Initialize semaphore empty, full and mutex.
2. Create two threads- producer thread and 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 critical section.
6. Call sem_post on mutex semaphore followed by full semaphore
7. before exiting 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 Code:

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

#define SIZE 3

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

sem_t full, empty;
pthread_mutex_t mutex;

void produce() {
    if(sem_trywait(&empty) != 0) {
        printf("Buffer is full!!\n");
        return;
    }

    pthread_mutex_lock(&mutex);
    item++;
    buffer[in] = item;
    printf("Producer produces the item %d\n", item);
    in = (in + 1) % SIZE;
    pthread_mutex_unlock(&mutex);
}
```

```

    sem_post(&full);
}

void consume() {
    if(sem_trywait(&full) != 0) {
        printf("Buffer is empty!!\n");
        return;
    }

    pthread_mutex_lock(&mutex);
    int consumed_item = buffer[out];
    printf("Consumer consumes item %d\n", consumed_item);
    out = (out + 1) % SIZE;
    pthread_mutex_unlock(&mutex);
    sem_post(&empty);
}

int main() {
    int choice;

    sem_init(&empty, 0, SIZE);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);

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

        switch(choice) {
            case 1: produce(); break;
            case 2: consume(); break;
            case 3: return 0;
            default: printf("Invalid choice!\n");
        }
    }

    return 0;
}

```

Sample Output:

```
1. Producer
2.Consumer
3.Exit
Enter your choice:1
Producer produces the item 1
Enter your choice:2
Consumer consumes item 1
Enter your choice:2 Buffer
is empty!!
Enter your choice:1
Producer produces the item 1
Enter your choice:1
Producer produces the
item 2 Enter your choice:1
Producer produces
the item 3 Enter your
choice:1 Buffer is full!!
Enter your choice:3
```

Output:

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

```
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

Buffer is full!!

1. Producer

2. Consumer

3. Exit

Enter your choice: 3

Result:

A solution for the producer consumer problem using semaphores has been implemented successfully and the output has been verified.