Ex. No.: 8
Date: 29/3/25

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 <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#define BUFFER_SIZE 3
int buffer[BUFFER_SIZE];
int count=0;
int item=1;
sem_t empty, full;
pthread_mutex_t mutex;
void *producer (void *arg){
    sem_wait(&empty);
    pthread_mutex_lock(&mutex);
    buffer[count] = item;
    printf("Producer produces item %d\n",item);
    item++;
    count++;
```

1. Producer
2. Consumer
3. Exit

Enter your choice : 1

Producer produces item 1

Enter your choice : 2
Consumer consumes item 1
~~Buffer is empty !!~~

Enter your choice : 2

~~Producer produces item~~
Buffer is empty !!

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 : 3

Exiting...

PDF Scanner ACE Scanner

```
    pthread_mutex_unlock(&mutex);
    sem_post(&full);
    return NULL;
}
void * consumer(void *arg){
    sem_wait(&full);
    pthread_mutex_lock(&mutex);
    int consumeditem = buffer[count-1];
    printf("Consumer consumed item %d\n", consumeditem);
    count--;
    pthread_mutex_unlock(&mutex);
    sem_post(&empty);
    return NULL;
}
int main(){
    pthread_t prodThread, consThread;
    int choice;
    sem_init(&empty, 0, BUFFER_SIZE);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);
    while(1){
        printf("\n1. Producer \n2. Consumer \n3. Exit \n Enter your choice:")
        scanf("%d", &choice);
        switch(choice){
            Case 1:
                pthread_create(&prodThread, NULL, producer, NULL);
                pthread_join(prodThread, NULL);
                break;
            Case 2:
                pthread_create(&consThread, NULL, consumer, NULL);
                pthread_join(consThread, NULL);
                break;
```


PDF Scanner ACE Scanner

```
case 3:
        printf("EXITING...\n");
        sem_destroy(&empty);
        sem_destroy(&full);
        pthread_mutex_destroy(&mutex);
        exit(0);

    default:
        printf("Invalid choice\n");
    }
}
return 0;
}
```

PDF Scanner ACE Scanner

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

Result:

Thus the program to implement solution to producer consumer problem using semaphores are succesfully executed.

ACE Scanner