Ex. No.: 8
Date: 28/3/25

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;
sem_t empty pty[1];
sem_t full;
P Thread _ mutex_t mutex.
void * producer (void * arg
{
    static int item = 1;
    P Thread _ mutex_ lock (& mutex);
```

```
if (count == BUFFER_SIZE)
{
    Print ("Buffer is full : \n");
    PThread_Mutex_unlock (& mutex).
    return NULL;
}
PThread_mutex_lock (& matrix);
sem_wait (& empty);
PThread_mutex_lock (& mutex).
Buffer [count] = item;
Printf ("Producer produces the item %d\n", item);
item ++;
count ++;
Pthread_mutex_unlock (& mutex);
sem_post (& full);
return NULL;
}
Void * Consumer (Void * arg)
    if (count == 0)
        printf ("Buffer is empty ! \n");
        return NULL;
    }
    sem_wait (& full);
    PThread_mutex_lock (& mutex);
```

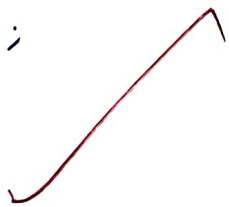
```
    if( count > 0) {
        int item = Buffer[count - 1];
        Printf(" Consumer consumes item %d \n", item);
        count = ;
    }.
  Pthread_mutex_unlock (& mutex);
  sem_post (& empty);
  return NULL;
}
int main()
{
   Pthread_t p_thread, cons_thread;
  int choice;
  sem_init (& full, 0, 0);
  P thread_mutex_init (& mutex, NULL);
while (1)
{
   printf("\n 1. produces \n 2. consumes \n 3. Exit \n Enter
                                        your choice:");
  Scanf ("%d, & choice);
  Switch (choice)
  {
      case 1:
          p thread_ creat (& prod thread, NULL, producer, NULL);
```

```
        pthread_join (pro_thread, NULL);
        break;

Case 2:
        pthread_create (&con_thread, NULL, Consumer, NULL);
        pthread_join (con_thread, NULL);
        Break;

Case 3:
        printf ("Exiting ...\n");
        sem_destroy (&empty);
        sem_destroy (&full);
        pthread_mutex_destroy (&mutex);
        exit (0);

default:
        printf ("Invalid choice!");
    }
}
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
Enter your choice: 2
Consumes some item 1
Enter your choice: 3
Exiting ....

Result:

Hence, the c program to implement solution to producer consumer problem using semaphores has been executed and written successful.