

**Ex. No. : 8**

**Date : 29.03.2025**

**Register No. : 230701385**

**Name : VISHWAK S**

---

## **PRODUCER CONSUMER USING SEMAPHORES**

### **Aim:**

To write a program to implement a 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 the critical section.
5. Produce/Consume the item in the critical section.
6. Call sem\_post on mutex semaphore followed by full semaphore
7. before exiting the 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:

```
D: > OS > C Prod_cons.c > producer(void *)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <pthread.h>
4  #include <semaphore.h>
5  #include <unistd.h>
6
7  #define BUFFER_SIZE 3
8
9  int buffer[BUFFER_SIZE];
10 int count = 0;
11
12 sem_t empty;
13 sem_t full;
14 pthread_mutex_t mutex;
15
16 void *producer(void *arg){
17     static int item = 1;
18
19     pthread_mutex_lock(&mutex);
20
21     if (count == BUFFER_SIZE)
22     {
23         printf("Buffer is full!!\n");
24         pthread_mutex_unlock(&mutex);
25         return NULL;
26     }
27
28     pthread_mutex_unlock(&mutex);
29     sem_wait(&empty);
30     pthread_mutex_lock(&mutex);
31
32     buffer[count] = item;
33     printf("Producer produces the item %d\n", item);
34     item++;
35     count++;
36
37     pthread_mutex_unlock(&mutex);
38     sem_post(&full);
39     return NULL;
40 }
41
```

```
41
42 void *consumer(void *arg)
43 {
44     if (count == 0)
45     {
46         printf("Buffer is empty!!\n");
47         return NULL;
48     }
49
50     sem_wait(&full);
51     pthread_mutex_lock(&mutex);
52
53     if (count > 0)
54     {
55         int item = buffer[count - 1];
56         printf("Consumer consumes item %d\n", item);
57         count--;
58     }
59
60     pthread_mutex_unlock(&mutex);
61     sem_post(&empty);
62
63     return NULL;
64 }
```

```

65
66 int main()
67 {
68     pthread_t prodThread, consThread;
69     int choice;
70
71     sem_init(&empty, 0, BUFFER_SIZE);
72     sem_init(&full, 0, 0);
73     pthread_mutex_init(&mutex, NULL);
74
75     while (1)
76     {
77         printf("\n1. Producer\n2. Consumer\n3. Exit\nEnter your choice: ");
78         scanf("%d", &choice);
79
80         switch (choice)
81         {
82             case 1:
83                 pthread_create(&prodThread, NULL, producer, NULL);
84                 pthread_join(prodThread, NULL);
85                 break;
86             case 2:
87                 pthread_create(&consThread, NULL, consumer, NULL);
88                 pthread_join(consThread, NULL);
89                 break;
90             case 3:
91                 printf("Exiting...\n");
92                 sem_destroy(&empty);
93                 sem_destroy(&full);
94                 pthread_mutex_destroy(&mutex);
95                 exit(0);
96             default:
97                 printf("Invalid choice! Please enter 1, 2, or 3.\n");
98         }
99     }
100
101     return 0;
102 }
103

```

## 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 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!!

1. Producer
2. Consumer
3. Exit
Enter your choice: 3
Exiting...
```

**Result:**

Hence a program to implement a solution to producer consumer problem using semaphores has been executed successfully.