Ex No:4 PRODUCER CONSUMER PROBLEM USING SEMAPHORE

Date : 04.03.2023

Aim:

To write a C Program that implements producer consumer Problem using semaphores.

Algorithm:

- Declare a structure to store BUFER and semaphores for synchronization
- Define the method for shared memory allocation
- Initialize structure pointer with shared memory
- Initialize semaphores

At Producer side:

- Semaphore down operation
- Place value to BUFFER
- Semaphore up operation

At Consumer Side:

- Semaphore down operation
- Semaphore for mutual exclusion
- Assign value of semphore full, to integer n
- Mutex up operation
- Semaphore up operation

CODING:

Problem.h

```
#include<stdio.h>
#include<semaphore.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<fcntl.h>
#define BUFFER_SIZE 10
#define PRODUCER_SLEEP_SEC 1
#define CONSUMER_SLEEP_SEC 1
#define KEY 1022
typedef struct
{
  int buff[BUFFER_SIZE];
  sem_t mutex, empty, full;
}MEM;
MEM *memory()
{
  key_t key = KEY;
  int shmid;
  shmid = shmget(key, sizeof(MEM),IPC_CREAT | 0666);
  return (MEM *) shmat(shmid,NULL,0);
}
void init()
{
 MEM *M=memory();
```

```
  sem_init(&M->mutex,1,1);
  sem_init(&M->empty,1,BUFFER_SIZE);
  sem_init(&M->full,1,0);
}
```

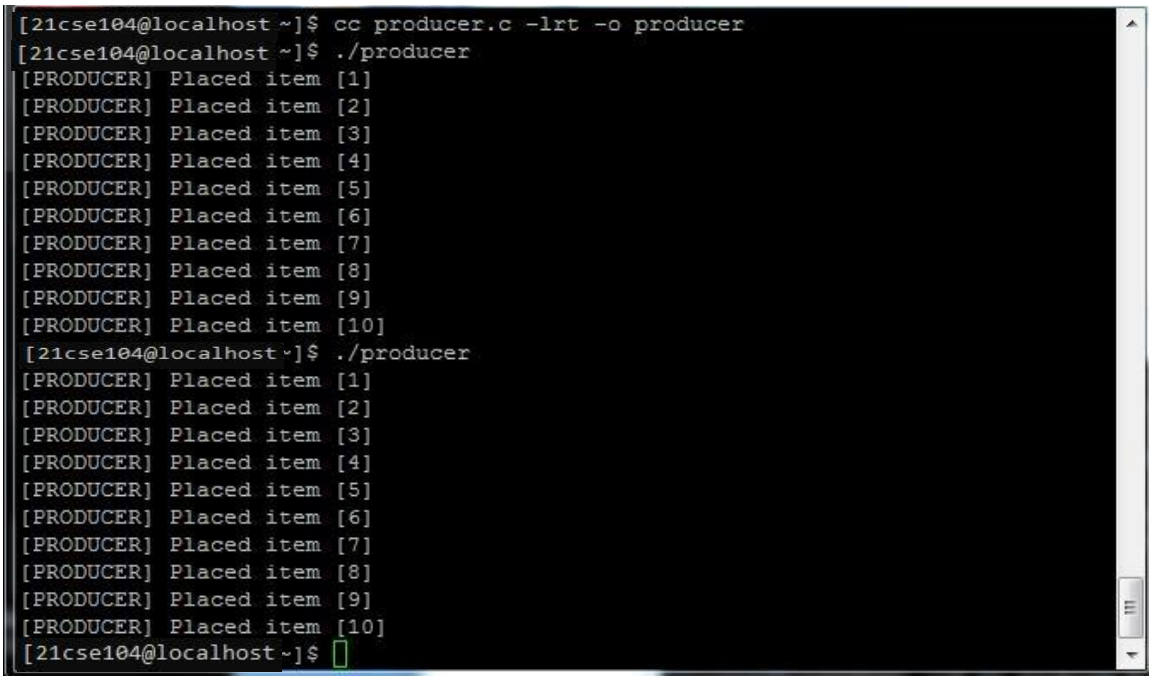
Producer.c

```
#include"problem.h"
void producer()
{
    int i=0,n;
    MEM *S = memory();
    while(i<10)
    {
        i++;
        sem_wait(&S->empty);
       sem_wait(&S->mutex);
       sem_getvalue(&S->full,&n);
       (S->buff)[n] = i;
        printf("[PRODUCER] Placed item [%d]\n", i);
        sem_post(&S->mutex);
        sem_post(&S->full);
        sleep(PRODUCER_SLEEP_SEC);
    }
}
main()
{
    init();
    producer();
}
```

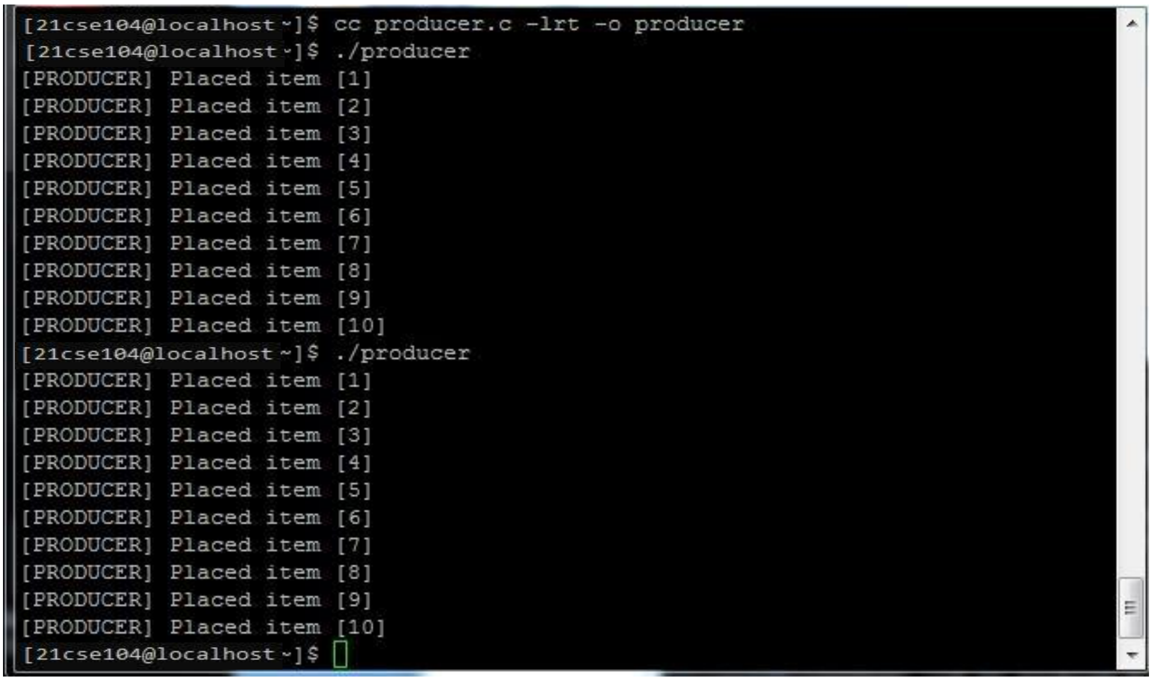
Consumer.c

```
#include"problem.h"
void consumer()
{
    int i=0,n;
    MEM *S = memory();
    while(i<10)
    {
        i++;
        sem_wait(&S->full);
        sem_wait(&S->mutex);
        sem_getvalue(&S->full,&n);
        printf("[CONSUMER] Removed item [%d]\n",(S->buff)[n]);
        sem_post(&S->mutex);
        sem_post(&S->empty);
        sleep(CONSUMER_SLEEP_SEC);
    }
}
main()
{
   consumer();
}
```

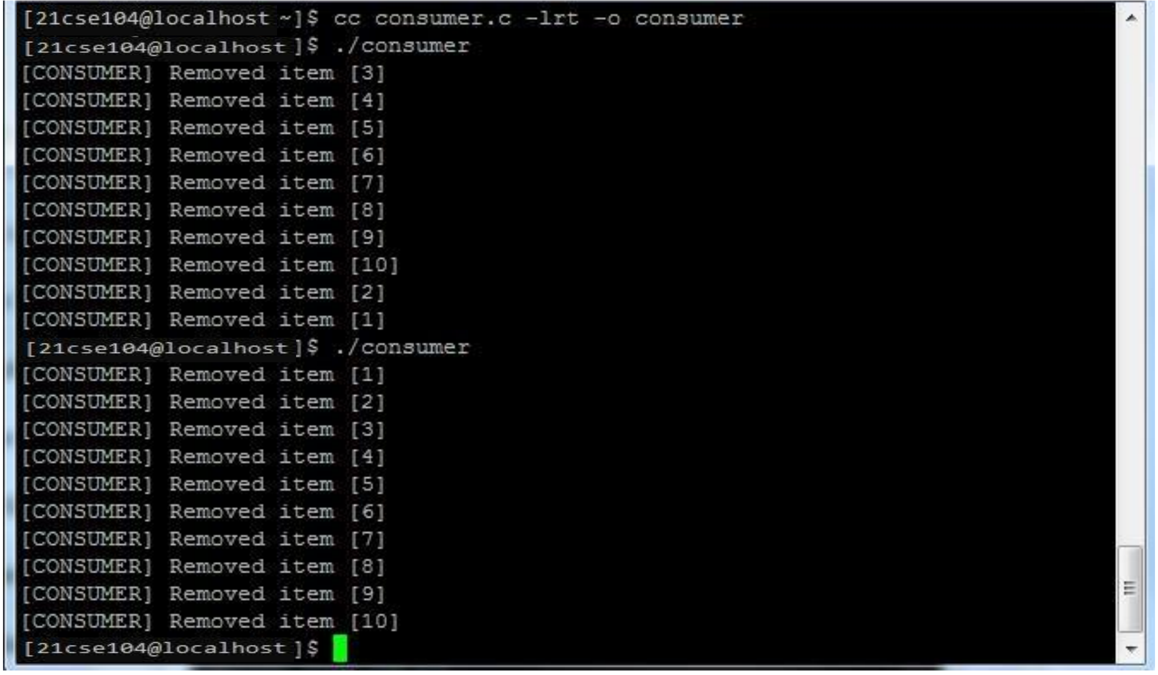
Output:



```
[21cse104@localhost ~]$ cc producer.c -lrt -o producer
[21cse104@localhost ~]$ ./producer
[PRODUCER] Placed item [1]
[PRODUCER] Placed item [2]
[PRODUCER] Placed item [3]
[PRODUCER] Placed item [4]
[PRODUCER] Placed item [5]
[PRODUCER] Placed item [6]
[PRODUCER] Placed item [7]
[PRODUCER] Placed item [8]
[PRODUCER] Placed item [9]
[PRODUCER] Placed item [10]
[21cse104@localhost ~]$ ./producer
[PRODUCER] Placed item [1]
[PRODUCER] Placed item [2]
[PRODUCER] Placed item [3]
[PRODUCER] Placed item [4]
[PRODUCER] Placed item [5]
[PRODUCER] Placed item [6]
[PRODUCER] Placed item [7]
[PRODUCER] Placed item [8]
[PRODUCER] Placed item [9]
[PRODUCER] Placed item [10]
[21cse104@localhost ~]$
```



```
[21cse104@localhost ~]$ cc producer.c -lrt -o producer
[21cse104@localhost ~]$ ./producer
[PRODUCER] Placed item [1]
[PRODUCER] Placed item [2]
[PRODUCER] Placed item [3]
[PRODUCER] Placed item [4]
[PRODUCER] Placed item [5]
[PRODUCER] Placed item [6]
[PRODUCER] Placed item [7]
[PRODUCER] Placed item [8]
[PRODUCER] Placed item [9]
[PRODUCER] Placed item [10]
[21cse104@localhost ~]$ ./producer
[PRODUCER] Placed item [1]
[PRODUCER] Placed item [2]
[PRODUCER] Placed item [3]
[PRODUCER] Placed item [4]
[PRODUCER] Placed item [5]
[PRODUCER] Placed item [6]
[PRODUCER] Placed item [7]
[PRODUCER] Placed item [8]
[PRODUCER] Placed item [9]
[PRODUCER] Placed item [10]
[21cse104@localhost ~]$
```



```
[21cse104@localhost ~]$ cc consumer.c -lrt -o consumer
[21cse104@localhost ]$ ./consumer
[CONSUMER] Removed item [3]
[CONSUMER] Removed item [4]
[CONSUMER] Removed item [5]
[CONSUMER] Removed item [6]
[CONSUMER] Removed item [7]
[CONSUMER] Removed item [8]
[CONSUMER] Removed item [9]
[CONSUMER] Removed item [10]
[CONSUMER] Removed item [2]
[CONSUMER] Removed item [1]
[21cse104@localhost ]$ ./consumer
[CONSUMER] Removed item [1]
[CONSUMER] Removed item [2]
[CONSUMER] Removed item [3]
[CONSUMER] Removed item [4]
[CONSUMER] Removed item [5]
[CONSUMER] Removed item [6]
[CONSUMER] Removed item [7]
[CONSUMER] Removed item [8]
[CONSUMER] Removed item [9]
[CONSUMER] Removed item [10]
[21cse104@localhost ]$
```

Observation	
Record	
Total	
Initial	

Result

Thus the implementation of producer consumer problem has been successfully created.