ASSIGINMENT NO. 4A

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h> //for system calls
#include <sys/types.h>
#include <sys/syscall.h> //to print thread id
#include <stdlib.h>
#define SIZE 3
void *producer(void *argp); //thread function for producer
void *consumer(void *argc); //thread function for consumer
struct Shared //structure
{
  int buff[SIZE];
  sem_t full, empty;
};
int front = -1, rear = -1;
pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER; //initialize mutex variable
struct Shared Sh;
int main()
{
  int prod, cons, i, j, k, l;
  pthread_t ptid, ctid; //producer and consumer thread ids
  sem_init(&Sh.empty, 0, 1); //initialize semaphore variable empty with value1..used in thread of single process
  sem_init(&Sh.full, 0, 0); //initialize semaphore variable full with value0..used in thread of single process
```

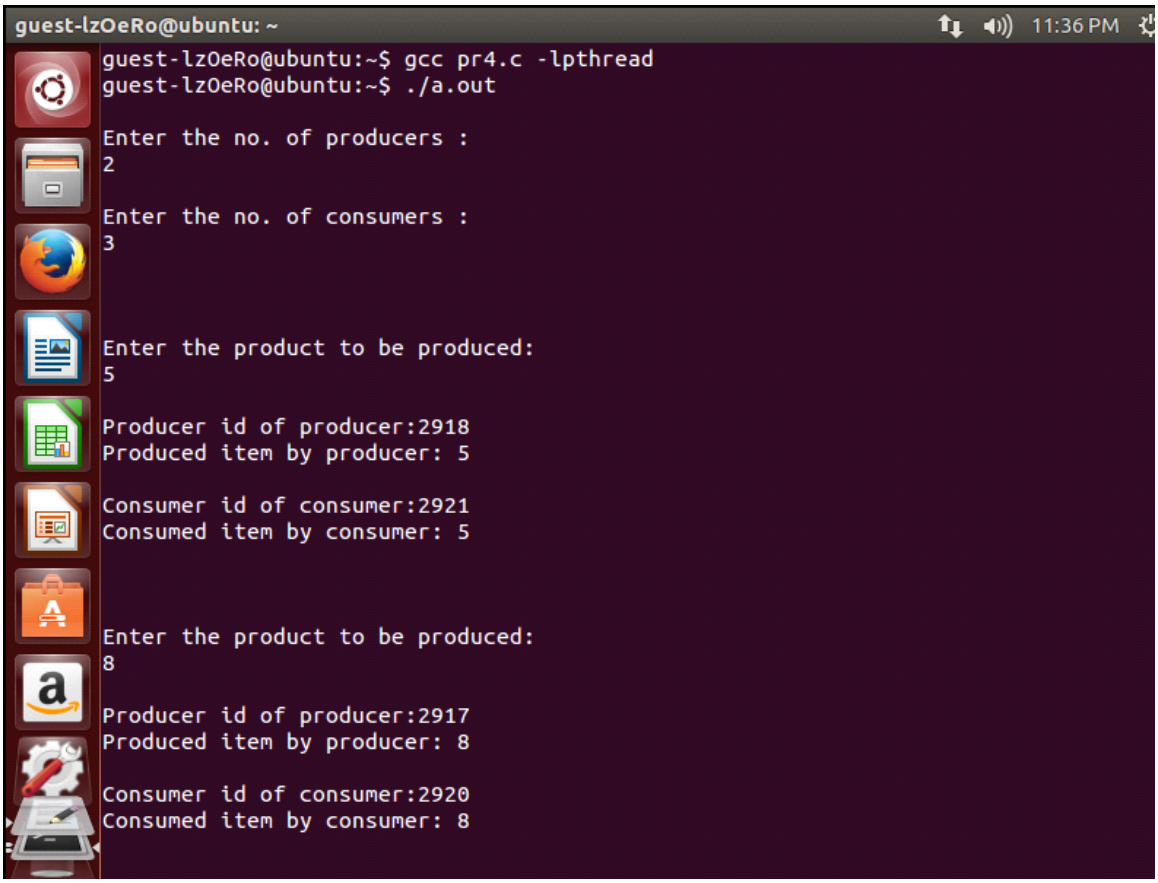
```
  printf("\nEnter the no. of producers :\n");
  scanf("%d", &prod);
  printf("\nEnter the no. of consumers :\n");
  scanf("%d", &cons);
  for (i = 0; i < prod; i++) //calling producer thread
  {
  pthread_create(&ptid, NULL, producer, NULL);
  }
  for (j = 0; j < cons; j++) //calling consumer thread
  {
  pthread_create(&ctid, NULL, consumer, NULL);
  }
  for (k = 0; k < prod; k++) //for joining producer thread
  {
  pthread_join(ptid, NULL);
  }
  for (l = 0; l < cons; l++) //for joining consumer thread
  {
  pthread_join(ctid, NULL);
  }
  return 0;
}
void *producer(void *argp) //producer function
{
  int i, item;
```

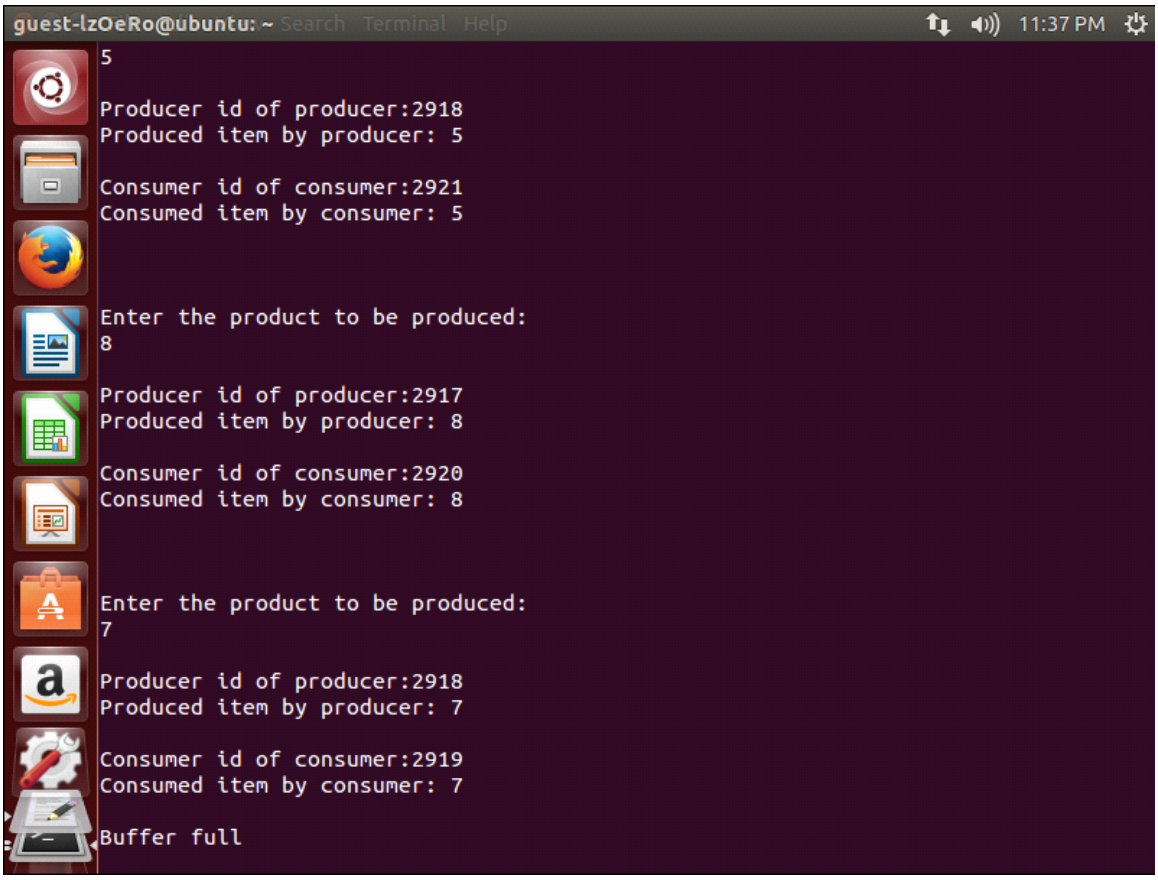
```
  while (1)
  {
  if (rear >= SIZE - 1) //if buffer is full
  {
  sleep(1);
  printf("\nBuffer full\n");
  exit(0);
  }
  else
  {
  if (front == -1) //for first element of bufffer
  {
  sem_wait(&Sh.empty); //critical section begins here
pthread_mutex_lock(&mut);
  sleep(3);
  printf("\n\n");
  printf("\nEnter the product to be produced:\n");
  scanf("%d", &item);
  Sh.buff[++rear] = item;
  printf("\nProducer id of producer:");
  printf("%ld\t", syscall(SYS_gettid));
  printf("\nProduced item by producer: %d\n", item);
  front = rear;
  pthread_mutex_unlock(&mut);
  sem_post(&Sh.full); //critical section ends here
```

```
  }
  else //for other elements
  {
  sem_wait(&Sh.empty); //critical section begins here
  pthread_mutex_lock(&mut);
  sleep(3);
  printf("\n\n");
  printf("\nEnter the product to be produced:\n");
  scanf("%d", &item);
  Sh.buff[++rear] = item;
  printf("\nProducer id of producer:");
  printf("%ld\t", syscall(SYS_gettid));
  printf("\nProduced item by producer: %d\n", item);
  pthread_mutex_unlock(&mut);
  sem_post(&Sh.full); //critical section ends here
  }
  }
  }
  return NULL;
  pthread_exit(0);
}
void *consumer(void *argc) //consumer function
{
  int i, item;
  while (1)
```

```
  {
  if (front == rear == -1) //if buffer is empty
  {
  printf("\nBuffer Empty..");
  break;
  }
  else
  {
  sem_wait(&Sh.full); //critical section begins here
  pthread_mutex_lock(&mut);
  item = Sh.buff[front++];
  printf("\nConsumer id of consumer:");
  printf("%ld\t", syscall(SYS_gettid));
  printf("\nConsumed item by consumer: %d\n", item);
  sem_post(&Sh.empty);
  pthread_mutex_unlock(&mut); //critical section ends here
  }
  }
  return NULL;
  pthread_exit(0);
}
```

OUTPUT:


guest-lzOeRo@ubuntu: ~
11:36 PM
guest-lzOeRo@ubuntu:~$ gcc pr4.c -lpthread
guest-lzOeRo@ubuntu:~$ ./a.out
Enter the no. of producers :
2
Enter the no. of consumers :
3
Enter the product to be produced:
5
Producer id of producer:2918
Produced item by producer: 5
Consumer id of consumer:2921
Consumed item by consumer: 5
Enter the product to be produced:
8
Producer id of producer:2917
Produced item by producer: 8
Consumer id of consumer:2920
Consumed item by consumer: 8


guest-lzOeRo@ubuntu: ~ Search Terminal Help
11:37 PM
5
Producer id of producer:2918
Produced item by producer: 5
Consumer id of consumer:2921
Consumed item by consumer: 5
Enter the product to be produced:
8
Producer id of producer:2917
Produced item by producer: 8
Consumer id of consumer:2920
Consumed item by consumer: 8
Enter the product to be produced:
7
Producer id of producer:2918
Produced item by producer: 7
Consumer id of consumer:2919
Consumed item by consumer: 7
Buffer full