Output

```
shmget: Success
Enter No. of Readers: 2
Enter No. of Writers: 1
Reader 0 is trying to Read
Reader 0 is Reading
Number of Readers = 1
Reader 1 is trying to Read
Writer 0 is trying to Write
Reader 0 is Leaving
Writer 0 is Writing
Data is now 0
Reader 1 is Reading
Writer 0 is Leaving
Number of Readers = 1
Reader 1 is Leaving
```

O/P Verified Sir.

```
Data read by the reader0 is 0
Data read by the reader2 is 0
Data read by the reader1 is 0
Data written by the writer0 is 1
Data written by the writer1 is 2
Data written by the writer2 is 3
```

Lab: 4

4/2/2020

READERS-WRITERS PROBLEM

Aim

To implement readers-writers problem using semaphore & shared memory.

Program

```
#include <stdio.h>
#include<pthread.h>
#include <semaphore.h>
sem_t mutex, writeblock
int data=0, rcount=0;
void *reader(void *arg)
{ int f;
  f=((int)arg);
  sem_wait(&mutex);
  rcount=rcount+1;
  if (rcount==1)
    sem_wait(&writeblock);
  sem_post(&mutex);
  printf("Data read by the reader %d is %d\n",f,data);
  sleep(1);
  sem_wait(&mutex);
  rcount=rcount-1;
  if (rcount==0)
    sem_post(&writeblock);
  sem_post(&mutex);
}
void *writer(void *arg)
{
  int f;
  f=((int)arg);
  sem_wait(&writeblock);
```

```
    data++;
    printf("Data written by the writer %d is %d\n", f, data);
    sleep(1);
    sem_post(&writeblock);
}

int main()
{
  int i,b;
  pthread_t rtid[5], wtid[5];
  sem_init(&mutex, 0, 1);
  sem_init(&writeblock, 0, 1);
  for(i=0; i<=2; i++)
  {
    pthread_create(&wtid[i], NULL, writer, (void *)i);
    pthread_create(&rtid[i], NULL, reader, (void *)i);
  }
  for(i=0; i<=2; i++)
  {
    pthread_join(wtid[i], NULL);
    pthread_join(rtid[i], NULL);
  }
  return 0;
}
```

Result

Implemented readers-writers problem using semaphore & shared memory.

30/30