College of Engineering,Trivandrum

Experiment 6

Network Programming Lab

Author:
Alan Anto

Registration Number :
TVE16CS09

February 20, 2019

Contents

1 Second Readers - Writers Problem

1.1 AIM

Implement the Second Readers-Writers problem

1.2 Theory

In second readers writers problem , no writer, once added to the queue, shall be kept waiting longer than absolutely necessary. This is called writers-preference .In this solution, preference is given to the writers. This is accomplished by forcing every reader to lock and release the readtry semaphore individually. The writers on the other hand don't need to lock it individually. Only the first writer will lock the readtry and then all subsequent writers can simply use the resource as it gets freed by the previous writer. The very last writer must release the readtry semaphore, thus opening the gate for readers to try reading.

1.3 Algorithm

```
int readcount, writecount;
semaphore rmutex, wmutex, readTry, resource;

//READER
reader() {
<ENTRY Section>
  readTry.P();
  rmutex.P();
  readcount++;
  if (readcount == 1)
    resource.P();
  rmutex.V();
  readTry.V();

<CRITICAL Section>
//reading is performed

<EXIT Section>
  rmutex.P();
```

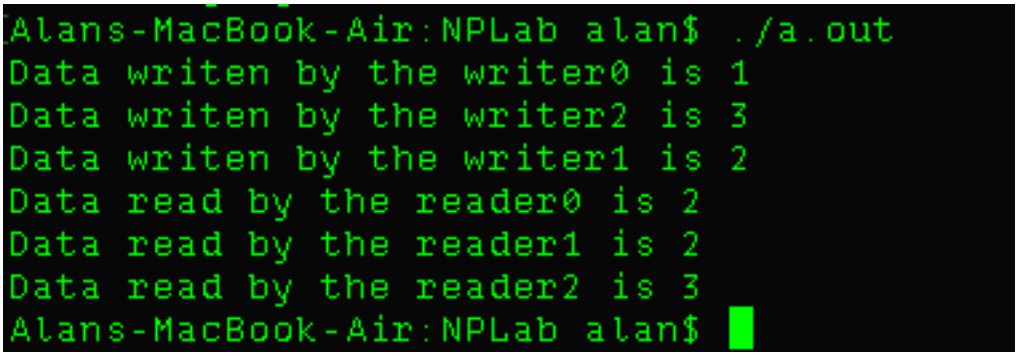
```
  readcount--;
  if (readcount == 0)
    resource.V();
  rmutex.V();
}

//WRITER
writer() {
<ENTRY Section>
  wmutex.P();
  writecount++;
  if (writecount == 1)
    readTry.P();
  wmutex.V();
<CRITICAL Section>
  resource.P();
  //writing is performed
  resource.V();

<EXIT Section>
  wmutex.P();
  if (writecount == 0)
    readTry.V();
  wmutex.V();
}
```

1.4 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 writen by the writer%d is %d\n",f,data);
    sleep(1);
    sem_post(&writeblock);
}

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);
    }
}
```

1.5 Output



1.6 Result

The second readers writers problem was implemented and the output was verified .