

ASSIGNMENT 4 (PART-B)

Problem Statement: Thread synchronization using mutex. Application to demonstrate: Reader-Writer problem with reader priority.

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include <unistd.h>

pthread_mutex_t mutex;
sem_t wrt;
int data = 0, rcount = 0;
void *reader(void *arg)
{
pthread_mutex_lock(&mutex);
rcount = rcount + 1;
if(rcount==1)
sem_wait(&wrt);
pthread_mutex_unlock(&mutex);
printf("Data read by the reader%d is %d\n",*((int *)arg),data);
sleep(1);
pthread_mutex_lock(&mutex);
rcount = rcount - 1;

if(rcount==0)
sem_post(&wrt);
pthread_mutex_unlock(&mutex);
}
void *writer(void *arg)
{
sem_wait(&wrt);
data++;
printf("Data written by the writer%d is %d\n",*((int *)arg),data);
sleep(1);
sem_post(&wrt);
}

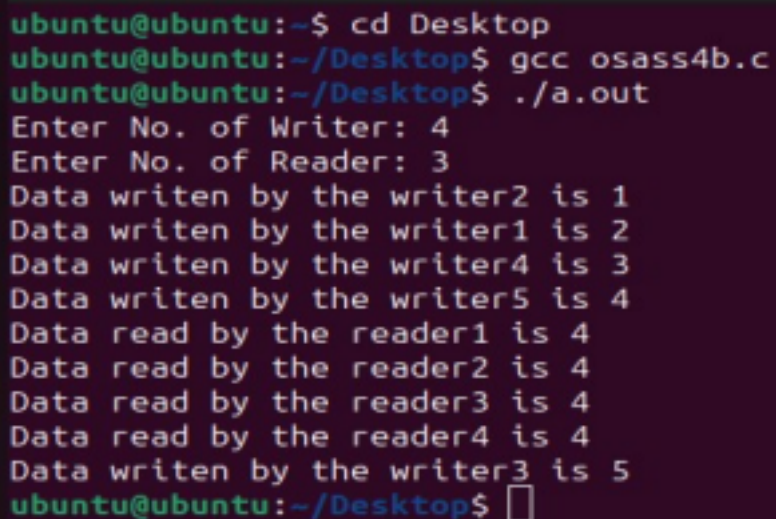
int main()
{
int wn,rn,b;
pthread_t rtid[5],wtid[5];
pthread_mutex_init(&mutex, NULL);
sem_init(&wrt,0,1);
printf("Enter No. of Writer: ");
```

NAME: Kaustubh Gajanan Indulkar
TE-IT-A 35026 (25027)

```
scanf("%d",&wn);
printf("Enter No. of Reader: ");
scanf("%d",&rn);
int arr[]={1,2,3,4,5,6,7,8,9,10};

for(int i=0;i<=wn;i++)
{
pthread_create(&wtid[i],NULL,writer,(void *)&arr[i]);
}
for(int i=0;i<=rn;i++)
{
pthread_create(&rtid[i],NULL,reader,(void *)&arr[i]);
}
for(int i=0;i<=wn;i++)
{
pthread_join(wtid[i],NULL);
}
for(int i=0;i<=rn;i++)
{
pthread_join(rtid[i],NULL);
}
return 0;
}
```

OUTPUT:



```
ubuntu@ubuntu:~$ cd Desktop
ubuntu@ubuntu:~/Desktop$ gcc osass4b.c
ubuntu@ubuntu:~/Desktop$ ./a.out
Enter No. of Writer: 4
Enter No. of Reader: 3
Data written by the writer2 is 1
Data written by the writer1 is 2
Data written by the writer4 is 3
Data written by the writer5 is 4
Data read by the reader1 is 4
Data read by the reader2 is 4
Data read by the reader3 is 4
Data read by the reader4 is 4
Data written by the writer3 is 5
ubuntu@ubuntu:~/Desktop$
```