**Reader Writer** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<stdio.h>

#include<sys/types.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

#define N 5

int b;

int readcount;

sem\_t mutex,wrt;

void\* w()

{

int i=1;

for(i=0;i<N;i++)

{

sem\_wait(&wrt); //wait

b=i;

i++;

//Writing is performed

printf("\nW :written =%d",b);

sem\_post(&wrt); //signal

printf("\nW going to sleep..");

sleep(2);

}

printf("Writer is living");

}

void \* r(void \* a)

{

int \*rr =(int \*) a ;

int i;

for(i=0;i<N;i++)

{

sem\_wait(&mutex);

readcount++;

if(readcount==1)

sem\_wait(&wrt);

sem\_post(&mutex);

//Readind is performed

printf("\nR :read=%d Reader no.=%d ",b,\*rr);

printf("\nReader%d:going to sleep...",\*rr);

sleep(2);

sem\_wait(&mutex);

readcount--;

if(readcount==0) //to check last reader

sem\_post(&wrt);

sem\_post(&mutex);

}

printf("Reader no.=%d living",\*rr);

}

void main()

{

pthread\_t w1,w2,r1,r2;

int rr1=1,rr2=2;

b=0;

readcount=0;

printf("\n value of shared variable is %d",b);

sem\_init(&(wrt),0,1);

sem\_init(&(mutex),0,1);

//Creating Threads

pthread\_create(&w1,NULL,w,NULL);

pthread\_create(&r1,NULL,r,(void \* )&rr1);

pthread\_create(&r2,NULL,r,(void \* )&rr2);

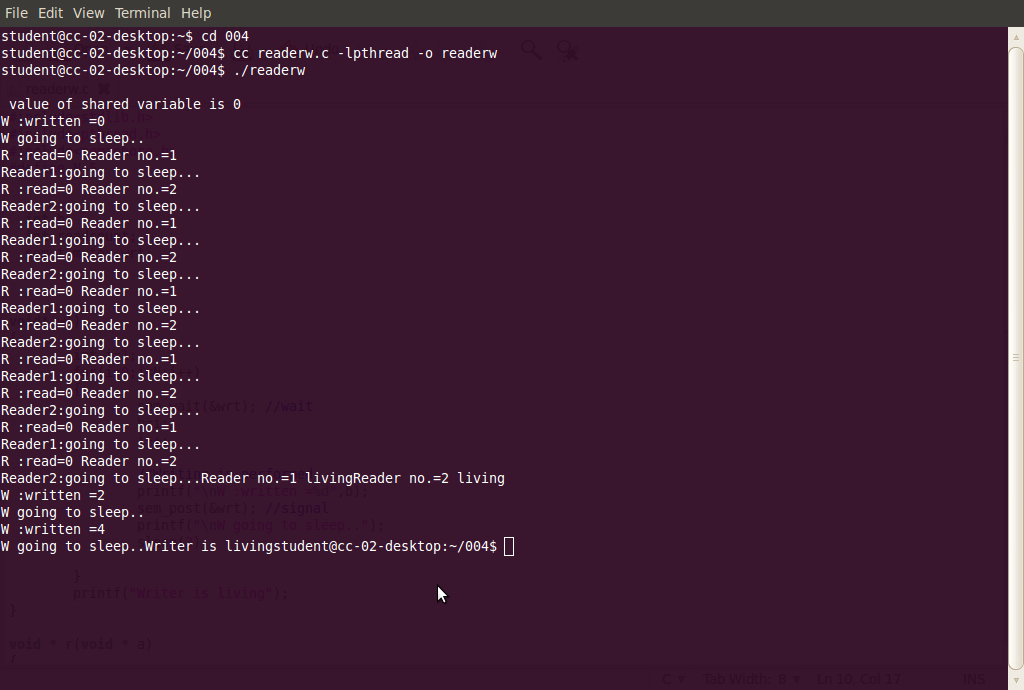
pthread\_join(w1,NULL);

pthread\_join(r1,NULL);

pthread\_join(r2,NULL);

}

Output :-



// To link semaphore – l rt and pthread.h – l pthread