**OS\_LAB\_10\_Assignment**

**CE\_054**

* Program :-

1. Priority Higher => Reader : Implementation of Code.

Code :-

// Author : Dhruv B Kakadiya

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define rdr 3

#define wtr 1

sem\_t rdrsema, wrtsema;

int rdrcnt = 0;

int content = 0;

void\* rdrFun(void \*args)

{

    int \*rdrArg = (int \*)args;

    while(1)

    {

        sem\_wait(&rdrsema);

        rdrcnt++;

        if(rdrcnt == 1)

        {

            sem\_wait(&wrtsema);

        }

        sem\_post(&rdrsema);

        printf("Reader %d is reading number %d\n", \*rdrArg, content);

        sem\_wait(&rdrsema);

        rdrcnt--;

        if(rdrcnt==0)

        {

            sem\_post(&wrtsema);

        }

        sem\_post(&rdrsema);

        sleep(2);

    }

}

void\* wrtFun(void \*args)

{

    int \*wrtArg=(int \*)args;

    int i = 0;

    while(1)

    {

        sem\_wait(&wrtsema);

        content++;

        printf("Writer number %d is writing number %d\n", \*wrtArg, content);

        sem\_post(&wrtsema);

        sleep(1);

    }

}

void main()

{

    int i=0;

    int rdrArr[rdr], wrtArr[wtr];

    pthread\_t rdr\_thread\_array[rdr], wtr\_thread\_array[wtr];

    sem\_init(&rdrsema, 0, 1);

    sem\_init(&wrtsema, 0, 1);

    for(i = 0 ; i < rdr ; i++)

    {

        rdrArr[i] = i + 1;

        pthread\_create(&rdr\_thread\_array[i], NULL, rdrFun, (void\*)&rdrArr[i]);

    }

    for(i = 0 ; i < wtr ; i++)

    {

        wrtArr[i] = i + 1;

        pthread\_create(&wtr\_thread\_array[i], NULL, wrtFun, (void\*)&wrtArr[i]);

    }

    for(i = 0 ; i < rdr ; i++)

    {

        pthread\_join(rdr\_thread\_array[i], NULL);

    }

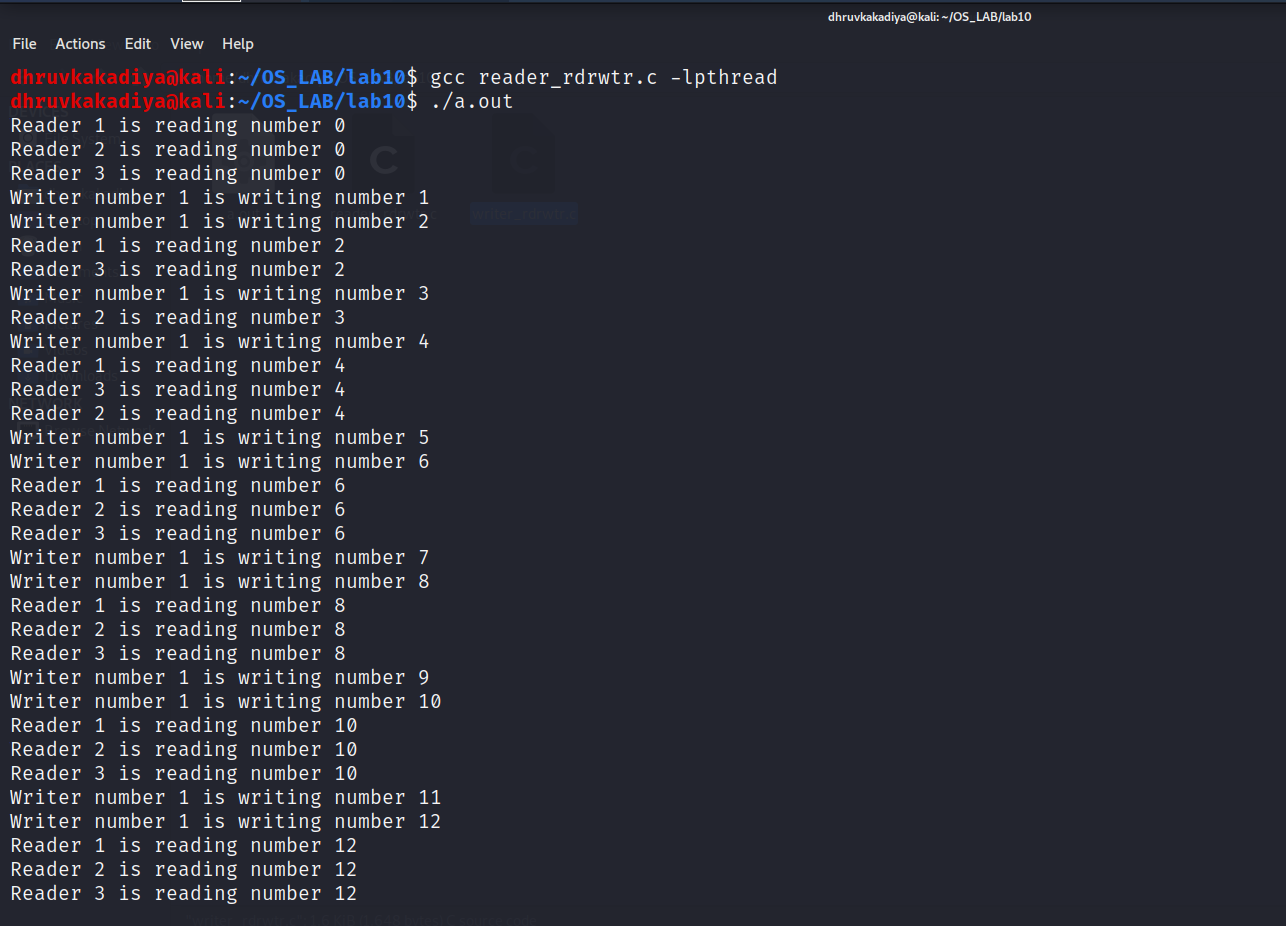
    for(i = 0 ; i < wtr ; i++)

    {

    }

}

Output :-



1. Higher priority => Writer : Implementation of Code :

Code :-

// Author : Dhruv B Kakadiya

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define rdr 3

#define wtr 2

sem\_t sem\_z, rdr\_sema\_cnt, wrt\_sema\_cnt, rdrsema, wrtsema;

int rdrCnt = 0, wrtCnt = 0;

int content = 0;

void \*rdr\_function(void\* args)

{

    int \*rdrArg = (int \*)args;

    while(1)

    {

        sem\_wait(&sem\_z);

        sem\_wait(&rdrsema);

        sem\_wait(&rdr\_sema\_cnt);

        rdrCnt++;

        if(rdrCnt == 1)

        {

            sem\_wait(&wrtsema);

        }

        sem\_post(&rdr\_sema\_cnt);

        sem\_post(&rdrsema);

        sem\_post(&sem\_z);

        printf("Reader %d is reading number %d\n", \*rdrArg, content);

        sem\_wait(&rdr\_sema\_cnt);

        rdrCnt--;

        if(rdrCnt==0)

        {

            sem\_post(&wrtsema);

        }

        sem\_post(&rdr\_sema\_cnt);

        sleep(3);

    }

}

void \*wrt\_function(void\* args)

{

    int \*wrtArg = (int \*)args;

    while(1)

    {

        sem\_wait(&wrt\_sema\_cnt);

        wrtCnt++;

        if(wrtCnt==1)

        {

            sem\_wait(&rdrsema);

        }

        sem\_post(&wrt\_sema\_cnt);

        sem\_wait(&wrtsema);

        content++;

        printf("Writer number %d is writing number %d\n",\*wrtArg, content);

        sem\_post(&wrtsema);

        sem\_wait(&wrt\_sema\_cnt);

        wrtCnt--;

        if(wrtCnt==0)

        {

            sem\_post(&rdrsema);

        }

        sem\_post(&wrt\_sema\_cnt);

        sleep(2);

    }

}

void main(){

    int i = 0;

    int rdrArr[rdr], wrtArr[wtr];

    pthread\_t rdr\_thread\_array[rdr], wrt\_thread\_array[wtr];

    sem\_init(&rdr\_sema\_cnt, 0, 1);

    sem\_init(&wrt\_sema\_cnt, 0, 1);

    sem\_init(&rdrsema, 0, 1);

    sem\_init(&wrtsema, 0, 1);

    sem\_init(&sem\_z, 0, 1);

    for(i = 0 ; i < rdr ; i++)

    {

        rdrArr[i] = i + 1;

        pthread\_create(&rdr\_thread\_array[i], NULL, rdr\_function, (void\*)&rdrArr[i]);

    }

    for(i = 0 ; i < wtr ; i++)

    {

        wrtArr[i] = i + 1;

        pthread\_create(&wrt\_thread\_array[i], NULL, wrt\_function, (void\*)&wrtArr[i]);

    }

    for(i = 0 ; i < rdr ; i++)

    {

        pthread\_join(rdr\_thread\_array[i], NULL);

    }

    for(i = 0 ; i < wtr ; i++)

    {

        pthread\_join(wrt\_thread\_array[i], NULL);

    }

}

Output :-

