//READER WRITER PROBLEM

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

#include<unistd.h>

void \*writer\_thr(int temp);// function prototypes

void \*reader\_thr(int temp);

sem\_t mutex;// declare semaphore variable

sem\_t wrt;

int readcount=0;//readcount is integer variable that keeps track of how many processes are currently reading the object.

int nwt,nrd;// nwt-->no of writer&nrd-->no of reader

void main()

{

long int i;

sem\_init(&mutex,0,1);// initialise semaphore variables

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

pthread\_t reader[100],writer[100];

//taking input

printf("\nEnter number of readers:");

scanf("%d",&nrd);

printf("\nEnter number of writers:");

scanf("%d",&nwt);

//creating threads

for(i=1;i<=nwt;i++)

{

pthread\_create(&writer[i],NULL,(void \*)writer\_thr,(int \*)i);

}

for(i=1;i<=nrd;i++)

{

pthread\_create(&reader[i],NULL,(void \*)reader\_thr,(int \*)i);

}

//joining thread

for(i=1;i<=nrd;i++)

{

pthread\_join(reader[i],NULL);

}

for(i=1;i<nwt;i++)

{

pthread\_join(writer[i],NULL);

}

sem\_destroy(&wrt);//destroying a semaphore that has been initialised by sem\_init.

sem\_destroy(&mutex);

}

void \*reader\_thr(int temp)

{

printf("\nReader %d is trying to enter database for reading.",temp);

sem\_wait(&mutex);

readcount++;// The number of readers has now increased by 1.

if(readcount==1)

sem\_wait(&wrt);//this ensure that no writer can enter if there is one reader.sem\_wait same like wait operation .

sem\_post(&mutex);//other readers can enter while this current reader in its critical section.

printf("\nReader %d is now reading in database.",temp);

sem\_wait(&mutex);

readcount--;//a reader wants to leave from the critical section.

if(readcount==0)//no reader is left in the critical section.

sem\_post(&wrt); //writers can enter now

sem\_post(&mutex);//readers leaves.

printf("\nReader %d has left the database.\n",temp);

sleep(3);//process suspend execution for a specified period

}

void \*writer\_thr(int temp)

{

printf("\nWriter %d is trying to enter database for modifying data",temp);

sem\_wait(&wrt);//writer request for critical section for entry

printf("\nWriter %d is writing in database.",temp);

sleep(3);

printf("\nWriter %d is leaving the database.\n",temp);

sem\_post(&wrt);//leave the critical section .sem\_post same as signal .

}