Câu 1:

//bai1

// C program to demonstrate working of Semaphores

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

sem\_t mutex1,mutex2;

void \*inle(void \*arg)

{

// wait

int i;

for(i=1;i<12;i+=2){

sem\_wait(&mutex2);

printf("Thred 1:%d\n",i);

sem\_post(&mutex1);

}

}

void \*inchan(void \*arg)

{

// wait

int i;

for(i=2;i<11;i+=2){

sem\_wait(&mutex1);

printf("Thred 2:%d\n",i);

sem\_post(&mutex2);

}

}

int main()

{

sem\_init(&mutex1, 0, 0);

sem\_init(&mutex2, 0, 1);

pthread\_t t1, t2;

pthread\_create(&t1, NULL, inle, NULL);

// sleep(2);

pthread\_create(&t2, NULL, inchan, NULL);

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

sem\_destroy(&mutex1);

sem\_destroy(&mutex2);

return 0;

}

Text

Description automatically generated

Câu 2:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#include <pthread.h>

long int total\_point;

void \*circle\_point(void \*param)

{

int \*pcount = (int \*)param;

int i;

for (i = 0; i < total\_point; i++)

{

double x = (double)rand() / (double)RAND\_MAX;

double y = (double)rand() / (double)RAND\_MAX;

double r = x \* x + y \* y;

if (r <= 1)

\*pcount = \*pcount + 1;

}

pthread\_exit(0);

}

int main(int argc, char const \*argv[])

{

if (argc != 2)

{

printf("Error\n");

return -1;

}

int NUM\_THREAD;

long int count\_circle = 0;

printf("Nhap so thread:");

scanf("%d", &NUM\_THREAD);

sleep(1);

pthread\_t tid[4] = {0};

int count[4] = {0};

total\_point = atoll(argv[1]) / NUM\_THREAD;

srand(time(NULL));

int i;

for (i = 0; i < NUM\_THREAD; i++)

pthread\_create(&tid[i], NULL, circle\_point, &count[i]);

for (i = 0; i < NUM\_THREAD; i++)

{

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

count\_circle += count[i];

}

double pi = 4.0 \* (double)count\_circle / (double)total\_point / (double)NUM\_THREAD;

printf("PI = %17.15f\n", pi);

return 0;

}

Text

Description automatically generated

Câu 3:

#include<stdio.h>

#include<semaphore.h>

#include<stdlib.h>

#include<pthread.h>

sem\_t m1, m2, m3;

void \*taosuon(void \*argv){

sem\_wait(&m1);

printf("Tao suon xe\n");

sem\_post(&m2);

}

void \*taobanh(void \*argv){

int i;

sem\_wait(&m2);

for(i=0;i<4;i++){

printf("Tao banh xe\n");

}

sem\_post(&m3);

}

void \*lapxe(void \*argv){

sem\_wait(&m3);

printf("Lap rap xe\n");

sem\_post(&m1);

}

void main()

{

int n,i;

printf("Nhap so luong xe: ");

scanf("%d", &n);

sleep(2);

for(i=0;i<n;i++){

sem\_init(&m1,0,1);

sem\_init(&m2,0,0);

sem\_init(&m3,0,0);

pthread\_t t1;

pthread\_t t2;

pthread\_t t3;

pthread\_create(&t1, NULL, taosuon, NULL);

pthread\_create(&t2, NULL, taobanh, NULL);

pthread\_create(&t3, NULL, lapxe, NULL);

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

pthread\_join(t3, NULL);

}

sem\_destroy(&m1);

sem\_destroy(&m2);

sem\_destroy(&m3);

}

Text

Description automatically generated

BÀI TẬP THÊM

Lab8.2

Bài 2:

#include <stdio.h>

#include <unistd.h>

#include <pthread.h>

#include <semaphore.h>

sem\_t mutex1, mutex2;

void\* W(void\* arg)

{

sem\_wait(&mutex1);

printf("Nguoi A toi.\n");

//critical section

sleep(1);

//Car is out

sem\_post(&mutex2);

printf("Nguoi A lui.\n");

}

void\* E(void\* arg)

{

sem\_wait(&mutex2);

printf("Nguoi B toi.\n");

//Car is out

sem\_post(&mutex1);

printf("Nguoi B lui.\n");

}

int main(void)

{

pthread\_t W1, W2, W3, W4, W5, E1, E2, E3, E4;

sem\_init(&mutex1, 0, 1);

sem\_init(&mutex2, 0, 0);

pthread\_create(&W1,NULL,W,NULL);

pthread\_create(&E1,NULL,E,NULL);

pthread\_create(&W2,NULL,W,NULL);

pthread\_create(&W3,NULL,W,NULL);

pthread\_create(&E2,NULL,E,NULL);

pthread\_create(&E3,NULL,E,NULL);

pthread\_create(&W4,NULL,W,NULL);

pthread\_create(&W5,NULL,W,NULL);

pthread\_create(&E4,NULL,E,NULL);

pthread\_join(W1,NULL);

pthread\_join(E1,NULL);

pthread\_join(W2,NULL);

pthread\_join(W3,NULL);

pthread\_join(E2,NULL);

pthread\_join(E3,NULL);

pthread\_join(W4,NULL);

pthread\_join(W5,NULL);

sem\_destroy(&mutex1);

sem\_destroy(&mutex2);

}

Text

Description automatically generated

Bài 8.3

#include <stdio.h>

#include <unistd.h>

#include <pthread.h>

#include <semaphore.h>

sem\_t mutex1, mutex2;

void\* W(void\* arg)

{

sem\_wait(&mutex1);

printf("Vermont toi.\n");

//critical section

sleep(1);

sem\_post(&mutex2);

printf("Vermont da qua cau.\n");

}

void\* E(void\* arg)

{

sem\_wait(&mutex2);

printf("Nguoi nguoi toi.\n");

sem\_post(&mutex1);

printf("Nguoi da qua cau.\n");

}

int main(void)

{

pthread\_t W1, W2, W3, W4, W5, E1, E2, E3, E4;

sem\_init(&mutex1, 0, 1);

sem\_init(&mutex2, 0, 0);

pthread\_create(&W1,NULL,W,NULL);

pthread\_create(&E1,NULL,E,NULL);

pthread\_join(W1,NULL);

pthread\_join(E1,NULL);

sem\_destroy(&mutex1);

sem\_destroy(&mutex2);

}

Text

Description automatically generated