

Tên	Đinh Phương My
MSSV	52100703
Nhóm thực hành	N101

# BÀI BÁO CÁO THỰC HÀNH LAB 8.2

## Mutex Clock.

**Câu 1:** Hãy thực hiện đồng bộ bằng semaphore, so sánh độ chính xác của số PI sau đó. Đồng thời đo thời gian chạy và cho biết sau khi đồng bộ, thời gian chạy đã tăng bao nhiêu %. Có thể cải tiến chương trình để giảm thiểu sự trả giá % thời gian chạy này không?

**Code:**

```

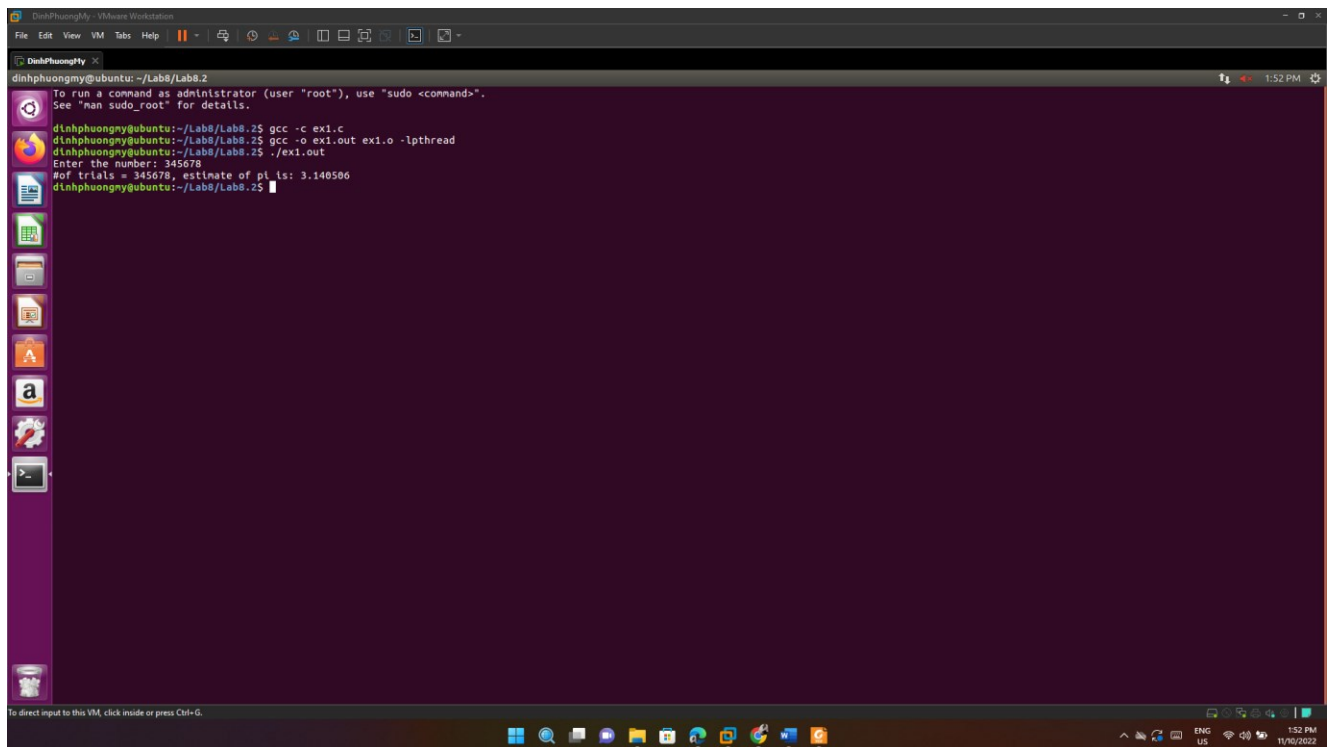
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <pthread.h>
#include <unistd.h>
#define SEED 35791246

int counter;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
struct arr
{
    int count;
    int niter;
};
void* count(void* arg)
{
    struct arr *ap = (struct arr*)arg;
    int i;
    double x, y, z;
    for(i = 0; i < (ap->niter)/2; i++)
    {
        pthread_mutex_lock(&mutex);
        x = (double) rand()/RAND_MAX;
        y = (double) rand()/RAND_MAX;
        z = x*x + y*y;
        if(z <= 1)
        {
            ap->count += 1;
        }
        pthread_mutex_unlock(&mutex);
    }
}

void main(int argc, char *argv)
{
    struct arr ar;
    ar.count = 0;
    printf("Enter the number: ");
    scanf("%d", &ar.niter);
    srand(SEED);
    pthread_t t[2];
    pthread_create(&t[1], NULL, count, (void*)&ar);
    pthread_create(&t[0], NULL, count, (void*)&ar);
    pthread_join(t[0], NULL);
    pthread_join(t[1], NULL);
    double pi = (double) ar.count/ar.niter*4;
    printf("For trials = %d, estimate of pi is: %f\n", ar.niter, pi);
}

```

## Run:

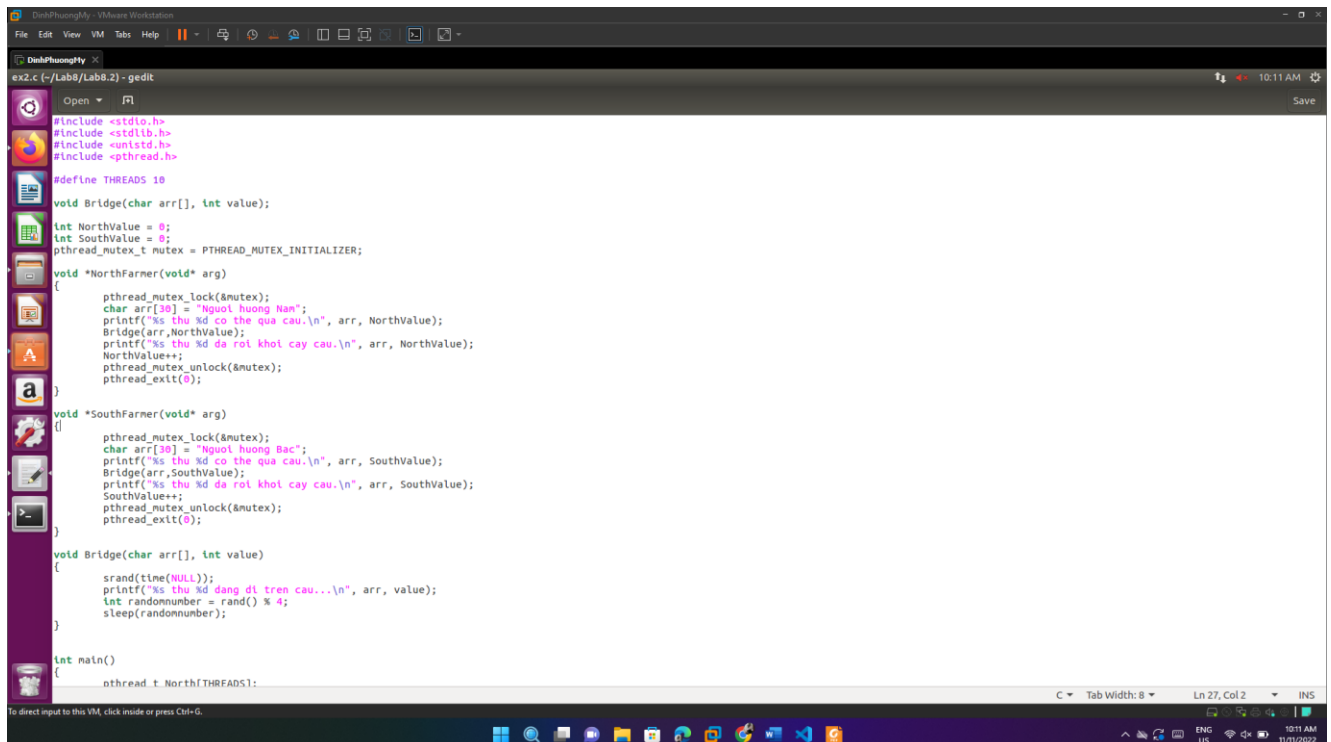


```
dinhphuongmy@ubuntu: ~/Lab8/Lab8.2
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -c ex1.c
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -o ex1.out ex1.o -lpthread
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ ./ex1.out
Enter the number: 345678
# of trials = 345678, estimate of pi is: 3.140586
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$
```

**Câu 2:** Xem xét bài toán “Nông dân qua cầu Vermont”, (tạm gọi bài tập này là phiên bản 2), chúng ta không kiểm soát tải trọng, mà giả định rằng người nông dân đến từ một trong hai hướng: Bắc và Nam, và không thể có 2 người cùng hướng đi lên cầu.

## Code:



```
ex2.c (-/Lab8/Lab8.2) - gedit
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

#define THREADS 10

void Bridge(char arr[], int value);

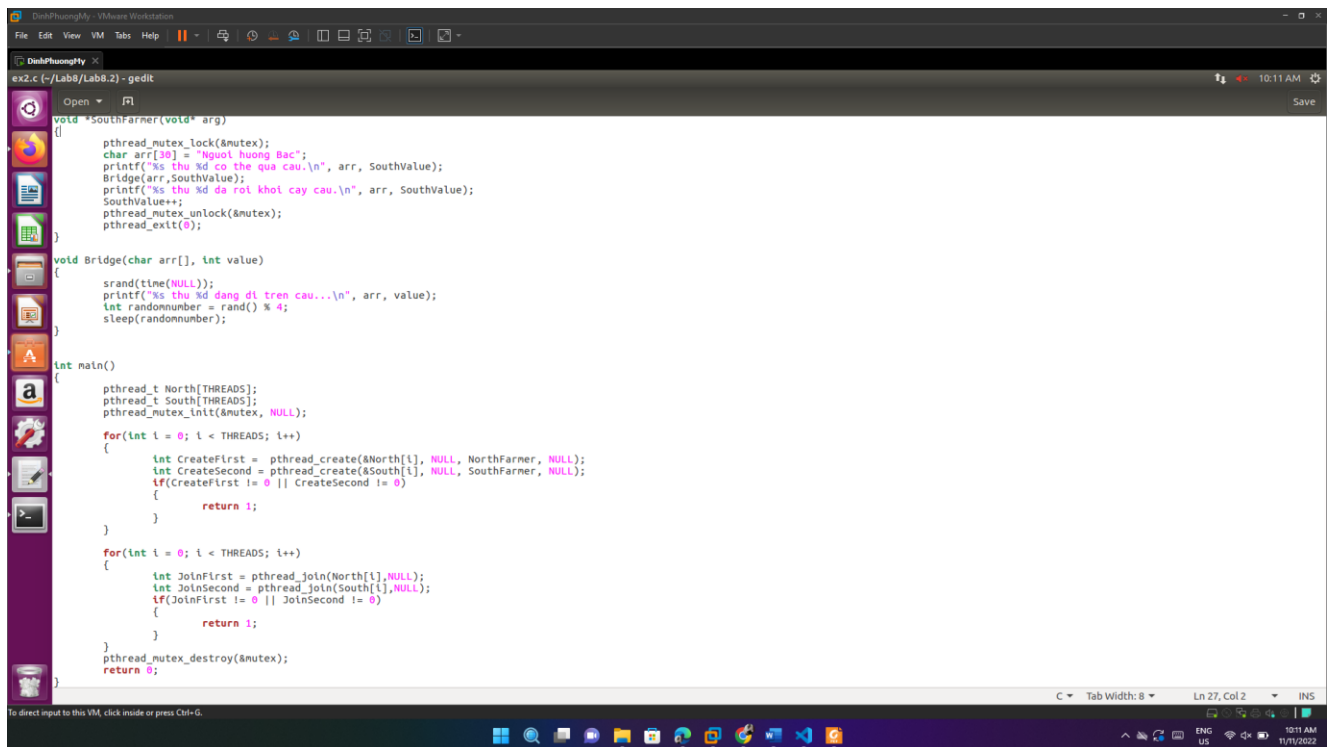
int NorthValue = 0;
int SouthValue = 0;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;

void *NorthFarmer(void* arg)
{
    pthread_mutex_lock(&mutex);
    char arr[30] = "Nguoi huong Nam";
    printf("%s thu xd co the qua cau.\n", arr, NorthValue);
    Bridge(arr, NorthValue);
    printf("%s thu xd da roi khoi cay cau.\n", arr, NorthValue);
    NorthValue++;
    pthread_mutex_unlock(&mutex);
    pthread_exit(0);
}

void *SouthFarmer(void* arg)
{
    pthread_mutex_lock(&mutex);
    char arr[30] = "Nguoi huong Bac";
    printf("%s thu xd co the qua cau.\n", arr, SouthValue);
    Bridge(arr, SouthValue);
    printf("%s thu xd da roi khoi cay cau.\n", arr, SouthValue);
    SouthValue++;
    pthread_mutex_unlock(&mutex);
    pthread_exit(0);
}

void Bridge(char arr[], int value)
{
    srand(time(NULL));
    printf("%s thu xd dang di tren cau...\n", arr, value);
    int randomnumber = rand() % 4;
    sleep(randomnumber);
}

int main()
{
    pthread_t North[THREADS];
```



```
void *SouthFarmer(void* arg)
{
    pthread_mutex_lock(&mutex);
    char arr[30] = "Nguoi huong Bac";
    printf("%s thu %d co the qua cau.\n", arr, SouthValue);
    Bridge(arr, SouthValue);
    printf("%s thu %d da roi khoi cay cau.\n", arr, SouthValue);
    SouthValue++;
    pthread_mutex_unlock(&mutex);
    pthread_exit(0);
}

void Bridge(char arr[], int value)
{
    srand(time(NULL));
    printf("%s thu %d dang di tren cau...\n", arr, value);
    int randomnumber = rand() % 4;
    sleep(randomnumber);
}

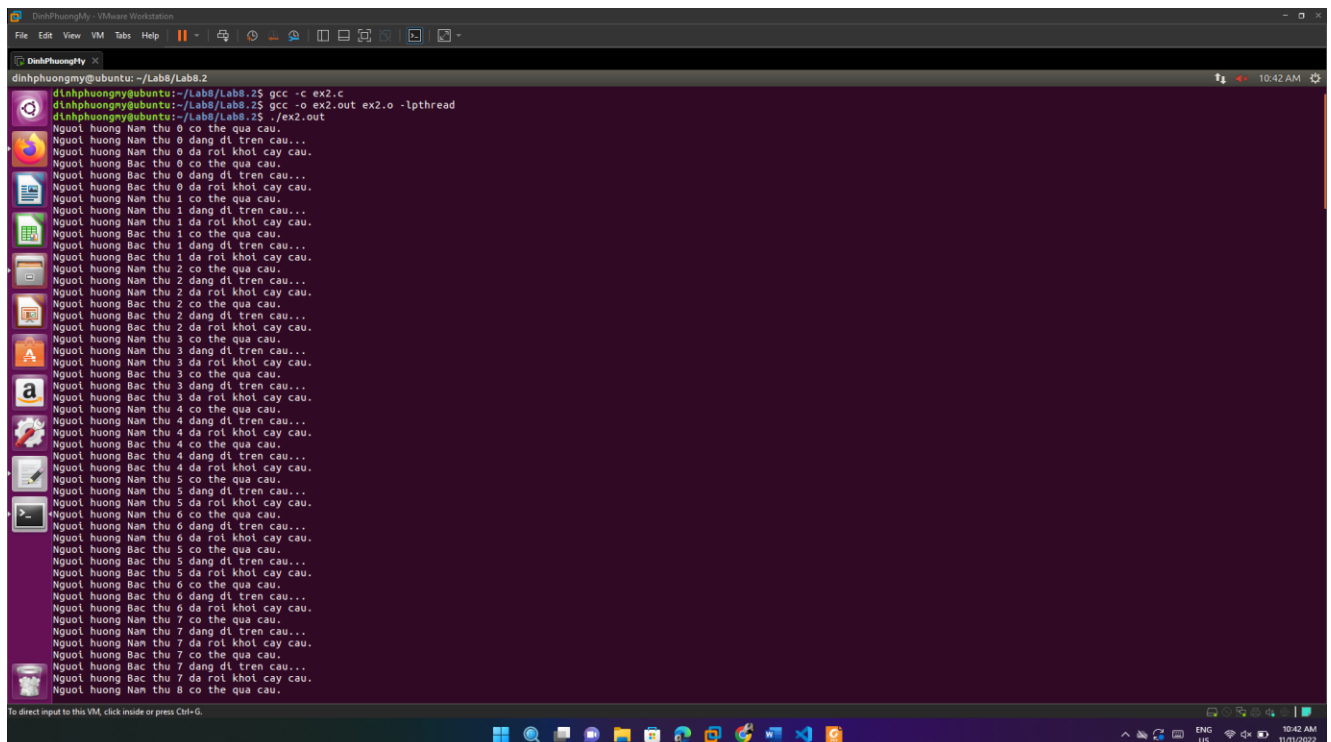
int main()
{
    pthread_t North[THREADS];
    pthread_t South[THREADS];
    pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;

    for(int i = 0; i < THREADS; i++)
    {
        int CreateFirst = pthread_create(&North[i], NULL, NorthFarmer, NULL);
        int CreateSecond = pthread_create(&South[i], NULL, SouthFarmer, NULL);
        if(CreateFirst != 0 || CreateSecond != 0)
        {
            return 1;
        }
    }

    for(int i = 0; i < THREADS; i++)
    {
        int JoinFirst = pthread_join(North[i], NULL);
        int JoinSecond = pthread_join(South[i], NULL);
        if(JoinFirst != 0 || JoinSecond != 0)
        {
            return 1;
        }
    }

    pthread_mutex_destroy(&mutex);
    return 0;
}
```

Run:

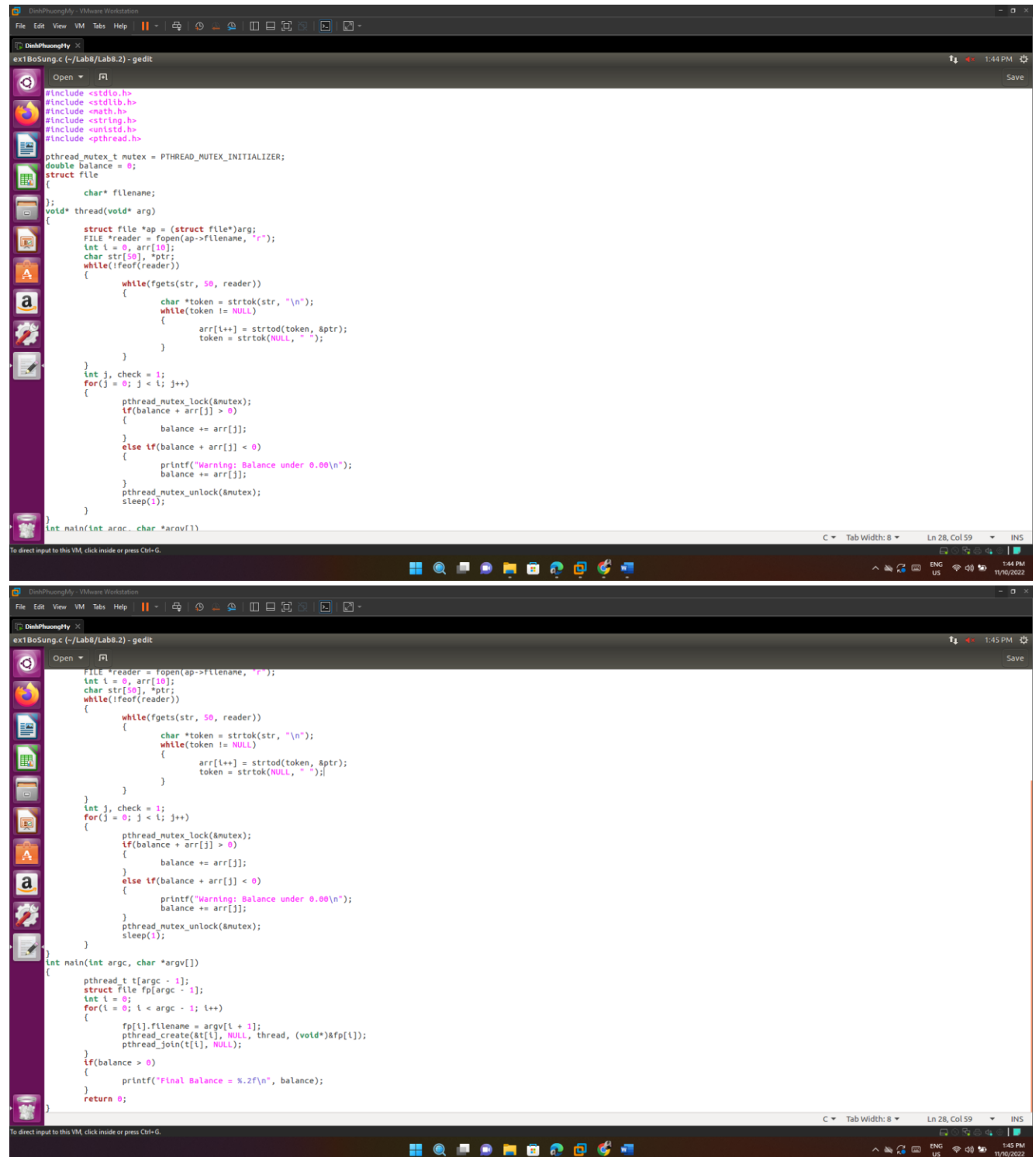


```
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -c ex2.c
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -o ex2.out ex2.o -lpthread
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ ./ex2.out
Nguoi huong Nam thu 0 co the qua cau.
Nguoi huong Nam thu 0 dang di tren cau...
Nguoi huong Nam thu 0 da roi khoi cay cau.
Nguoi huong Bac thu 0 co the qua cau.
Nguoi huong Bac thu 0 dang di tren cau...
Nguoi huong Bac thu 0 da roi khoi cay cau.
Nguoi huong Nam thu 1 co the qua cau.
Nguoi huong Nam thu 1 dang di tren cau...
Nguoi huong Bac thu 1 co the qua cau.
Nguoi huong Bac thu 1 dang di tren cau...
Nguoi huong Bac thu 1 da roi khoi cay cau.
Nguoi huong Nam thu 2 co the qua cau.
Nguoi huong Nam thu 2 dang di tren cau...
Nguoi huong Bac thu 2 co the qua cau.
Nguoi huong Bac thu 2 dang di tren cau...
Nguoi huong Bac thu 2 da roi khoi cay cau.
Nguoi huong Nam thu 3 co the qua cau.
Nguoi huong Nam thu 3 dang di tren cau...
Nguoi huong Bac thu 3 da roi khoi cay cau.
Nguoi huong Bac thu 3 co the qua cau.
Nguoi huong Bac thu 3 dang di tren cau...
Nguoi huong Nam thu 4 co the qua cau.
Nguoi huong Nam thu 4 dang di tren cau...
Nguoi huong Bac thu 4 da roi khoi cay cau.
Nguoi huong Bac thu 4 co the qua cau.
Nguoi huong Bac thu 4 dang di tren cau...
Nguoi huong Bac thu 4 da roi khoi cay cau.
Nguoi huong Nam thu 5 co the qua cau.
Nguoi huong Nam thu 5 dang di tren cau...
Nguoi huong Bac thu 5 da roi khoi cay cau.
Nguoi huong Bac thu 5 co the qua cau.
Nguoi huong Bac thu 5 dang di tren cau...
Nguoi huong Bac thu 5 da roi khoi cay cau.
Nguoi huong Nam thu 6 co the qua cau.
Nguoi huong Nam thu 6 dang di tren cau...
Nguoi huong Nam thu 6 da roi khoi cay cau.
Nguoi huong Bac thu 6 co the qua cau.
Nguoi huong Bac thu 6 dang di tren cau...
Nguoi huong Nam thu 7 dang di tren cau...
Nguoi huong Bac thu 7 da roi khoi cay cau.
Nguoi huong Bac thu 7 dang di tren cau...
Nguoi huong Bac thu 7 da roi khoi cay cau.
Nguoi huong Nam thu 8 co the qua cau.
```

**Bài tổng hợp số 1:** Tình trạng cạnh tranh (race condition) có thể xuất hiện trong nhiều hệ thống máy tính. Hãy xem xét một hệ thống ngân hàng duy trì số dư tài khoản với hai hàm thực thi: deposit(số tiền) và withdraw(số tiền). Hai hàm này được truyền vào số tiền sẽ được gửi hoặc rút từ số dư tài khoản ngân hàng. Giả sử rằng người chồng và người vợ chia sẻ một tài khoản ngân

hàng. Một cách đồng thời, người chồng gọi hàm `withdraw()` và người vợ gọi hàm `deposit()`. Mô tả làm thế nào một tình trạng cạnh tranh có thể xảy đến và làm cách nào để ngăn chặn tình trạng cạnh tranh này xảy ra.

## Code:



```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
double balance = 0;

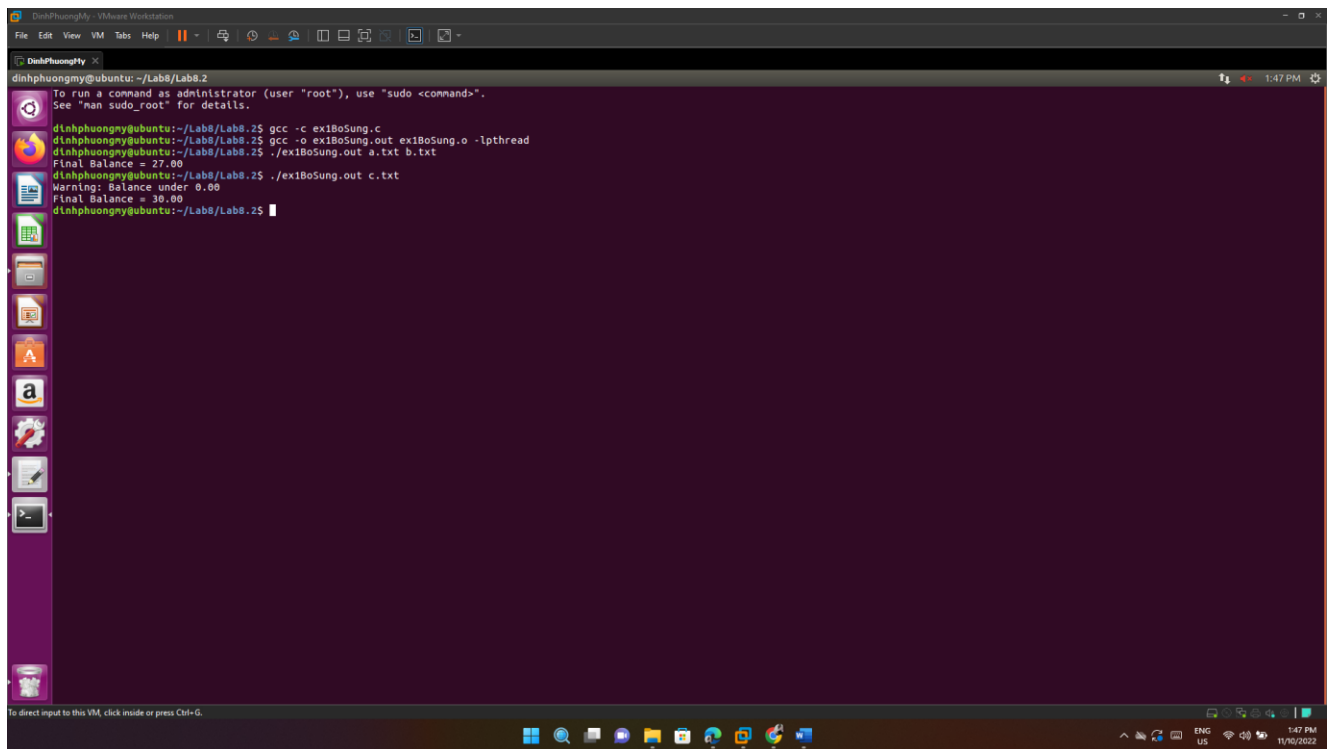
struct file
{
    char* filename;
};

void* thread(void* arg)
{
    struct file *ap = (struct file*)arg;
    FILE *reader = fopen(ap->filename, "r");
    int i = 0, arr[10];
    char str[50], *ptr;
    while(!feof(reader))
    {
        while(fgets(str, 50, reader))
        {
            char *token = strtok(str, "\n");
            while(token != NULL)
            {
                arr[i++] = strtod(token, &ptr);
                token = strtok(NULL, " ");
            }
        }
        int j, check = 1;
        for(j = 0; j < i; j++)
        {
            pthread_mutex_lock(&mutex);
            if(balance + arr[j] > 0)
            {
                balance += arr[j];
            }
            else if(balance + arr[j] < 0)
            {
                printf("Warning: Balance under 0.00\n");
                balance += arr[j];
            }
            pthread_mutex_unlock(&mutex);
            sleep(1);
        }
    }
}

int main(int argc, char *argv[])
{
    FILE *reader = fopen(argv[1], "r");
    int i = 0, arr[10];
    char str[50], *ptr;
    while(!feof(reader))
    {
        while(fgets(str, 50, reader))
        {
            char *token = strtok(str, "\n");
            while(token != NULL)
            {
                arr[i++] = strtod(token, &ptr);
                token = strtok(NULL, " ");
            }
        }
        int j, check = 1;
        for(j = 0; j < i; j++)
        {
            pthread_mutex_lock(&mutex);
            if(balance + arr[j] > 0)
            {
                balance += arr[j];
            }
            else if(balance + arr[j] < 0)
            {
                printf("Warning: Balance under 0.00\n");
                balance += arr[j];
            }
            pthread_mutex_unlock(&mutex);
            sleep(1);
        }
    }
}

int main(int argc, char *argv[])
{
    pthread_t t[argc - 1];
    struct file fp[argc - 1];
    int i = 0;
    for(i = 0; i < argc - 1; i++)
    {
        fp[i].filename = argv[i + 1];
        pthread_create(&t[i], NULL, thread, (void*)&fp[i]);
        pthread_join(t[i], NULL);
    }
    if(balance > 0)
    {
        printf("Final Balance = %.2f\n", balance);
    }
    return 0;
}
```

# Run:



```
DinhPhuongMy - VMware Workstation
File Edit View VM Tabs Help
DinhPhuongMy x
dinhphuongmy@ubuntu: ~/Lab8/Lab8.2
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -c ex1BoSung.c
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ gcc -o ex1BoSung.out ex1BoSung.o -lpthread
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ ./ex1BoSung.out a.txt b.txt
Final Balance = 27.00
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$ ./ex1BoSung.out c.txt
Warning: Balance under 0.00
Final Balance = 30.00
dinhphuongmy@ubuntu:~/Lab8/Lab8.2$
```

To direct input to this VM, click inside or press Ctrl+G.

1:47 PM 11/10/2022