ASSIGNMENT # 4

NAME : SHAHERYAR ASHFAQ

ROLL NO : **20P-0128**

SECTION : BS-CS **4-B**

SUBJECT : OPERATING SYSTEM

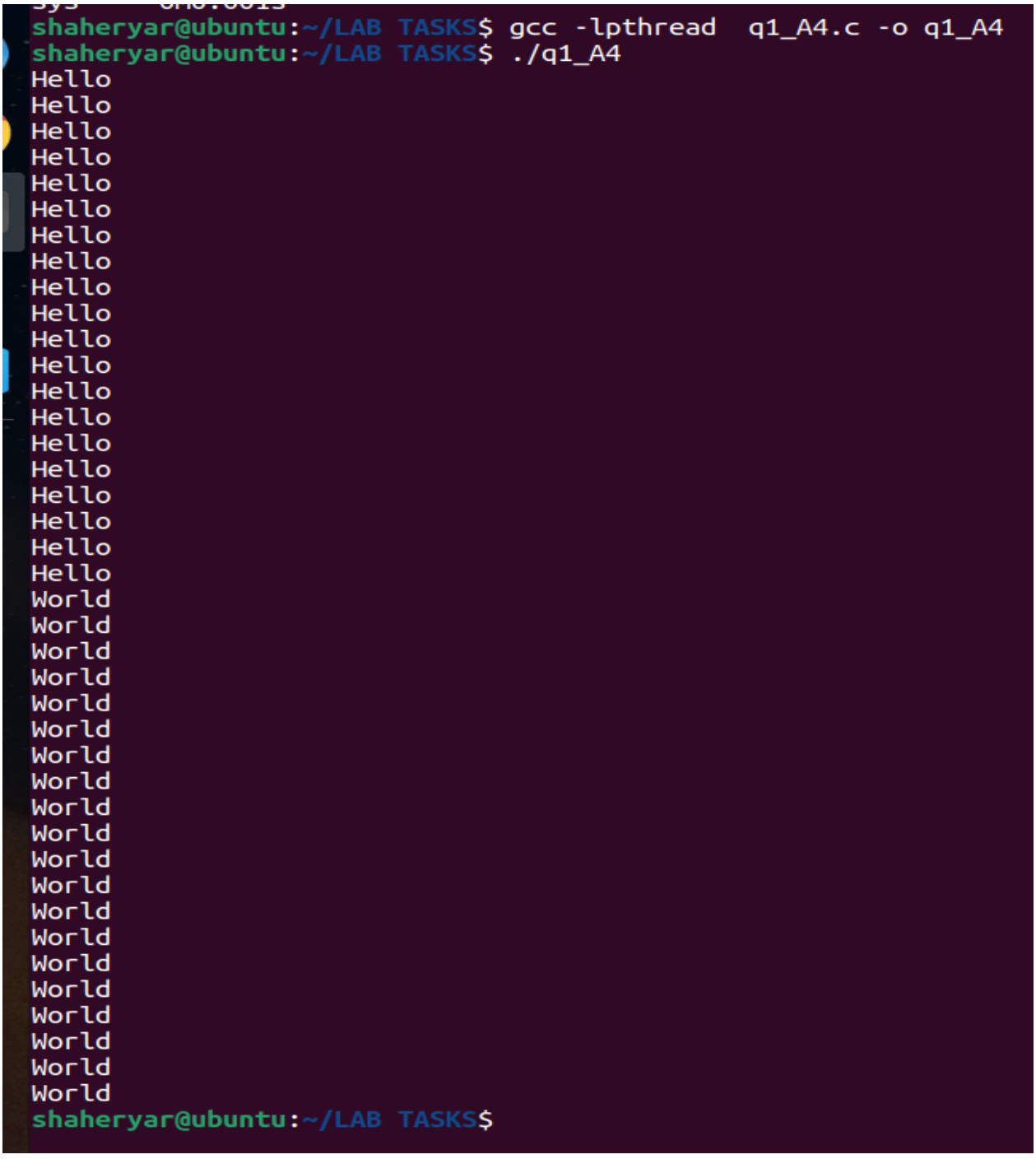
QUESTION 1

C q1_A4.c > main()

```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void * thread1(){

    for(int c=0 ; c<10 ; c++){
        printf("Hello\n");
    }
}
void * thread2(){

    for(int c=0 ; c<10 ; c++){
        printf("World\n");
    }
}
int main(){
    int status;
    pthread_t tid1,tid2,tid3,tid4;

    pthread_create(&tid1,NULL,thread1,NULL);
 pthread_create(&tid3,NULL,thread1,NULL);

    pthread_join(tid1,NULL);


    pthread_join(tid3,NULL);
    pthread_create(&tid2,NULL,thread2,NULL);

    pthread_create(&tid4,NULL,thread2,NULL);

    pthread_join(tid4,NULL);
     pthread_join(tid2,NULL);

    return 0;

}
```

OUTPUT WITH 2 THREADS

OUTPUT WITH 4 THREADS



OUTPUT EXPLANATION

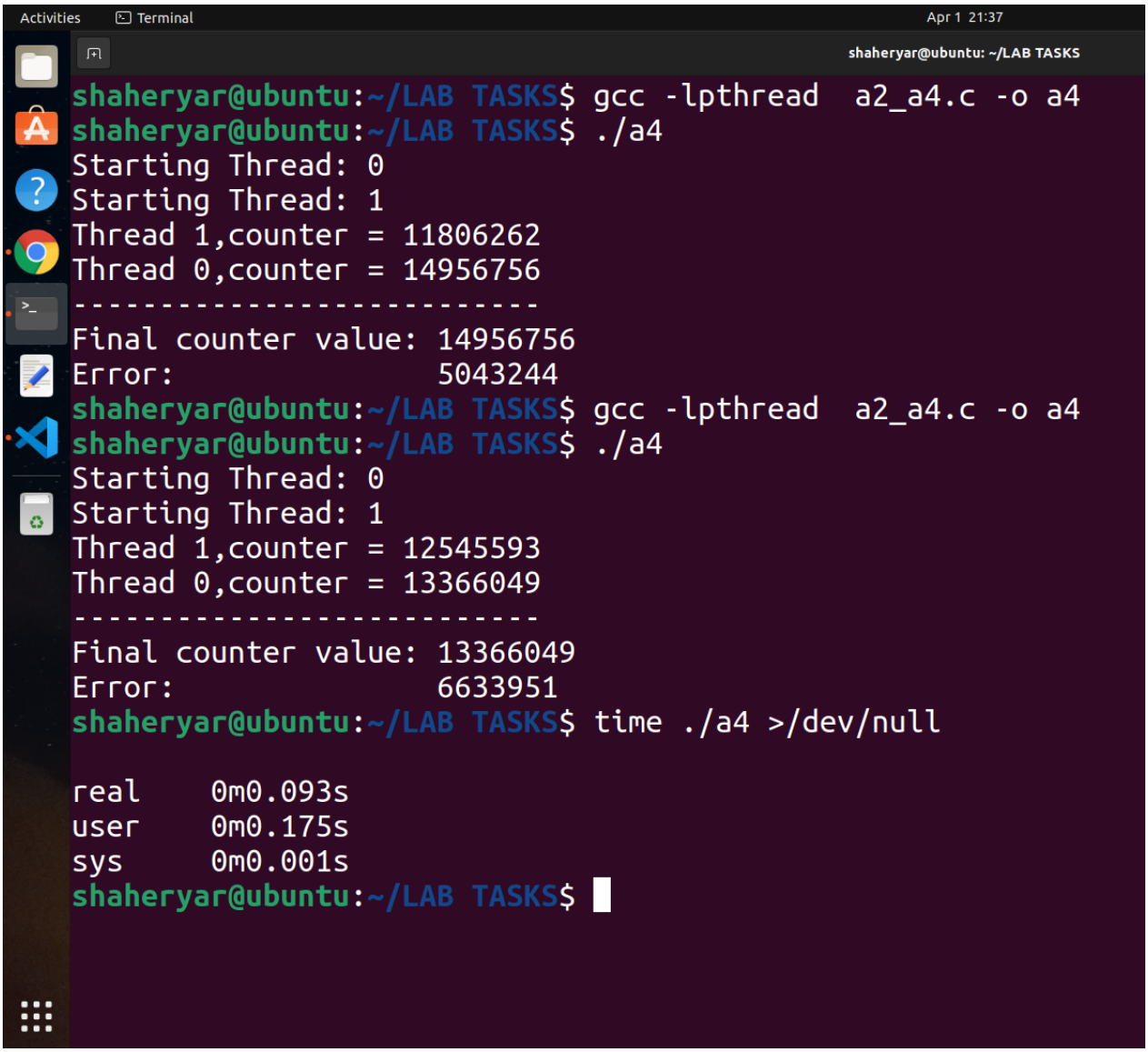
The main thread waits till the newly created thread exits. Therefore, the final line of the output is printed only after the new thread exits.

The thread scan terminate independently of each other by not using the pthread_join function.

q1_A4.c | a2_a4.c | test.c

a2_a4.c > main()

```c
#include <unistd.h>
#include <sys/types.h>
#include <errno.h>
#include <stdlib.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>

#define NUM_RUNS 10000000

void handler (void *ptr);

int counter;

int main(){
    int i[2];
    pthread_t thread_a;
    pthread_t thread_b;

    i[0] = 0;
    i[1] = 1;

    pthread_create ( &thread_a, NULL, (void *) &handler, (void *) &i[0]);
    pthread_create ( &thread_b, NULL, (void *) &handler, (void *) &i[1]);

    pthread_join( thread_a, NULL);
    pthread_join( thread_b, NULL);

    printf("---------------------------\n");
    printf("Final counter value: %d\n", counter);
    printf("Error:               %d\n", (NUM_RUNS*2-counter));
    exit(0);

}

void handler(void *ptr){
    int iter=0;
    int thread_num;
    thread_num=*((int*)ptr);
    printf("Starting Thread: %d \n",thread_num);

    while(iter<NUM_RUNS){
        counter++;
        iter +=1;
    }
    printf("Thread %d,counter = %d \n",thread_num,counter);
    pthread_exit(0);
}
```



Q.1 What should be the value of the counter variable at the end?

Ans 14956756

Q.2 What is the value you get?

Thread 0 Counter = 14956756

Thread 1 Counter = 11806262

Q.3 How large is the error and how much does it vary on different runs?

Ans: Error Value = 5043244.
Variance = 1,590,707

Q.4 How much user time (roughly) does the program take to run on your system?

Ans : 0.175s