# National Textile University

# Department of Computer Science

## Name

Gulam Rasool

## Reg No:

23-NTU-CS-FL-1159

## Semester:

5th

## Lab:

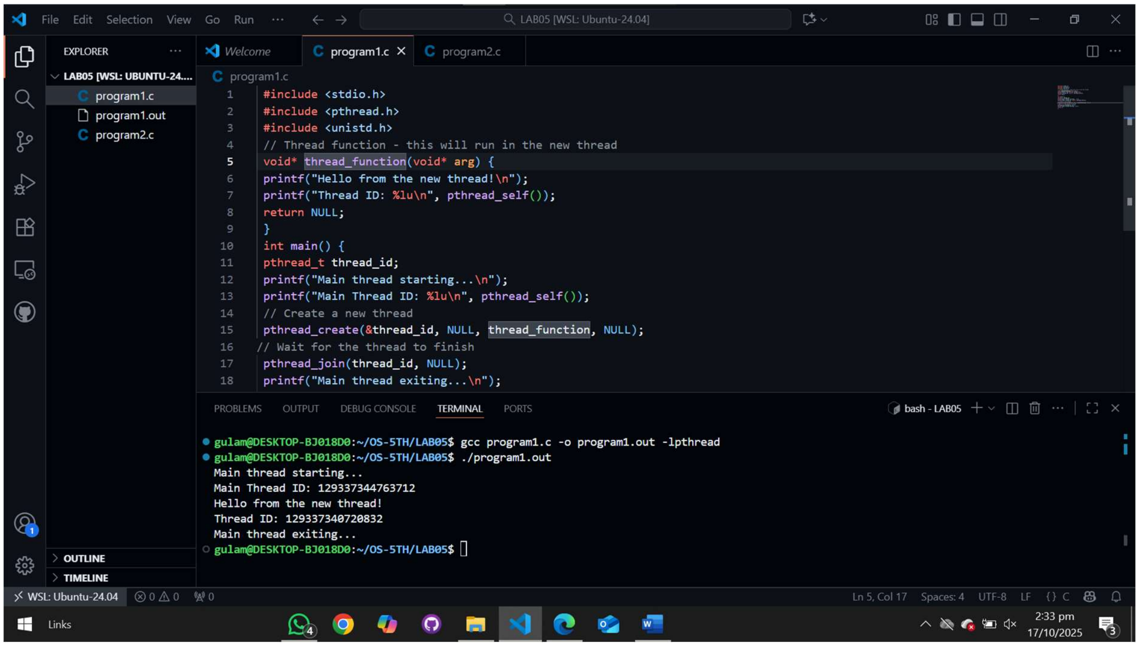
05-Class_Task

## Submitted to:

Sir Nasir Mehmood

# Lab-05(Class_Task)

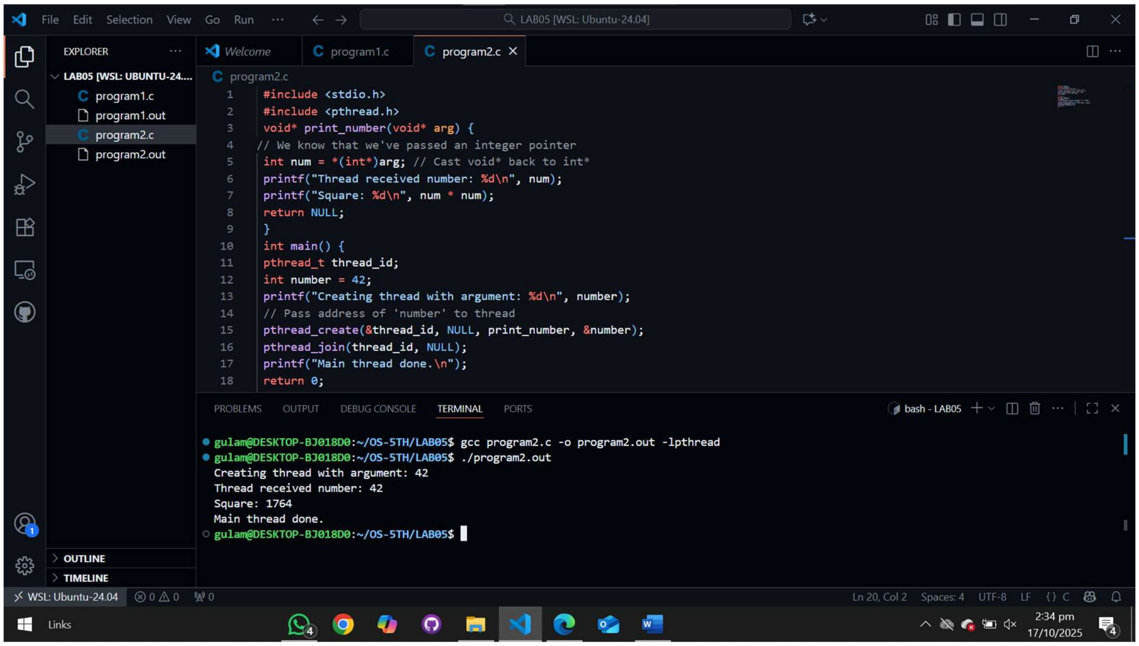
## Program 1: Creating a Simple Thread


File Edit Selection View Go Run
LAB05 [WSL: Ubuntu-24.04]
EXPLORER
LAB05 [WSL: UBUNTU-24....
program1.c
program1.out
program2.c
Welcome
program1.c
program2.c
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - LAB05
OUTLINE
TIMELINE
WSL: Ubuntu-24.04
Ln 5, Col 17 Spaces: 4 UTF-8 LF {} C
Links
2:33 pm
17/10/2025

```c
 #include <stdio.h>
 #include <pthread.h>
 #include <unistd.h>
 // Thread function - this will run in the new thread
 void* thread_function(void* arg) {
 printf("Hello from the new thread!\n");
 printf("Thread ID: %lu\n", pthread_self());
 return NULL;
 }
 int main() {
 pthread_t thread_id;
 printf("Main thread starting...\n");
 printf("Main Thread ID: %lu\n", pthread_self());
 // Create a new thread
 pthread_create(&thread_id, NULL, thread_function, NULL);
// Wait for the thread to finish
 pthread_join(thread_id, NULL);
 printf("Main thread exiting...\n");
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program1.c -o program1.out -lpthread
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program1.out
Main thread starting...
Main Thread ID: 129337344763712
Hello from the new thread!
Thread ID: 129337340720832
Main thread exiting...
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

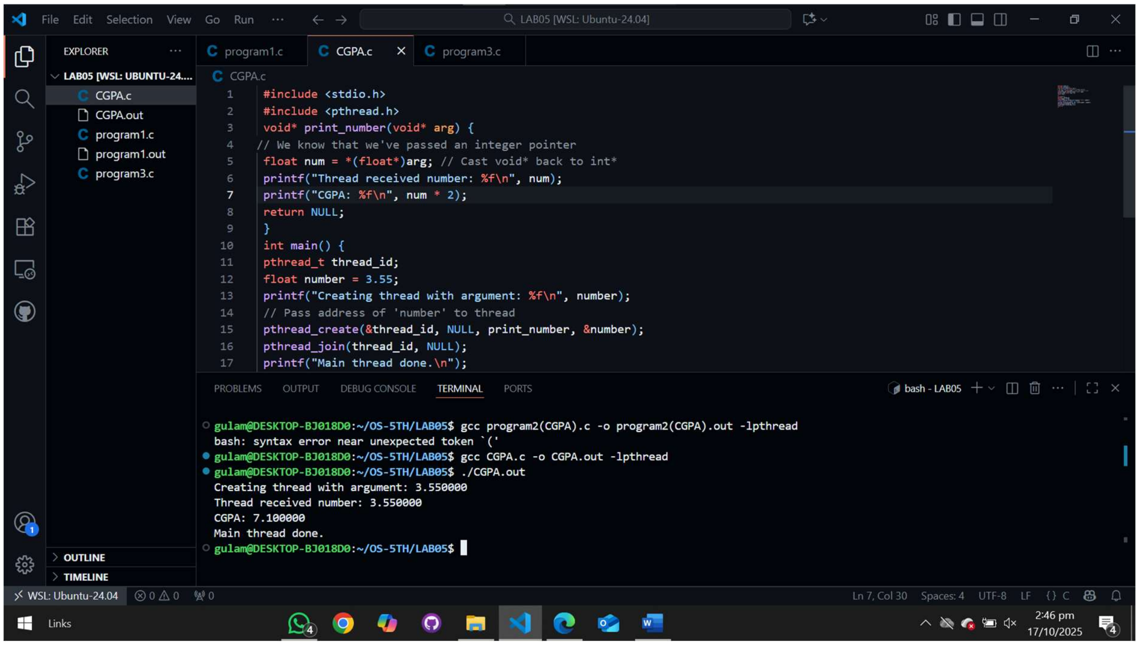
## Program 2: Passing Arguments to Threads


File Edit Selection View Go Run
LAB05 [WSL: Ubuntu-24.04]
EXPLORER
LAB05 [WSL: UBUNTU-24....
program1.c
program1.out
program2.c
program2.out
Welcome
program1.c
program2.c
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - LAB05
OUTLINE
TIMELINE
WSL: Ubuntu-24.04
Ln 20, Col 2 Spaces: 4 UTF-8 LF {} C
Links
2:34 pm
17/10/2025

```c
 #include <stdio.h>
 #include <pthread.h>
 void* print_number(void* arg) {
// We know that we've passed an integer pointer
 int num = *(int*)arg; // Cast void* back to int*
 printf("Thread received number: %d\n", num);
 printf("Square: %d\n", num * num);
 return NULL;
 }
 int main() {
 pthread_t thread_id;
 int number = 42;
 printf("Creating thread with argument: %d\n", number);
 // Pass address of 'number' to thread
 pthread_create(&thread_id, NULL, print_number, &number);
 pthread_join(thread_id, NULL);
 printf("Main thread done.\n");
 return 0;
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program2.c -o program2.out -lpthread
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program2.out
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

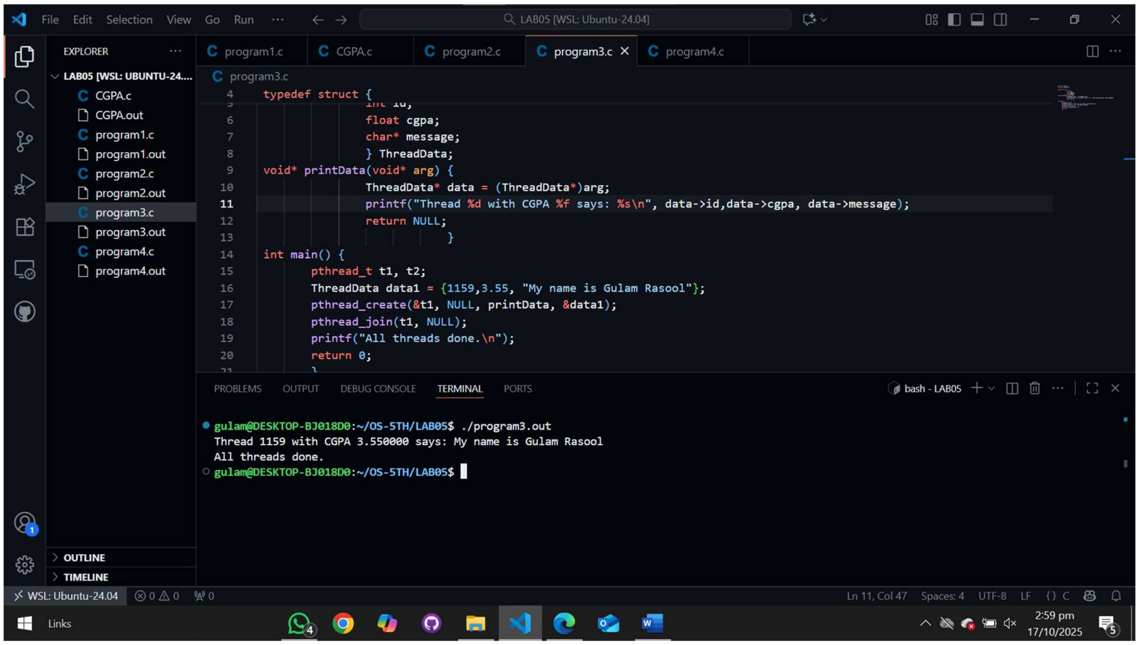
# Program 2: Double CGPA



```c
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
// We know that we've passed an integer pointer
float num = *(float*)arg; // Cast void* back to int*
printf("Thread received number: %f\n", num);
printf("CGPA: %f\n", num * 2);
return NULL;
}
int main() {
pthread_t thread_id;
float number = 3.55;
printf("Creating thread with argument: %f\n", number);
// Pass address of 'number' to thread
pthread_create(&thread_id, NULL, print_number, &number);
pthread_join(thread_id, NULL);
printf("Main thread done.\n");
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program2(CGPA).c -o program2(CGPA).out -lpthread
bash: syntax error near unexpected token `('
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc CGPA.c -o CGPA.out -lpthread
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./CGPA.out
Creating thread with argument: 3.550000
Thread received number: 3.550000
CGPA: 7.100000
Main thread done.
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

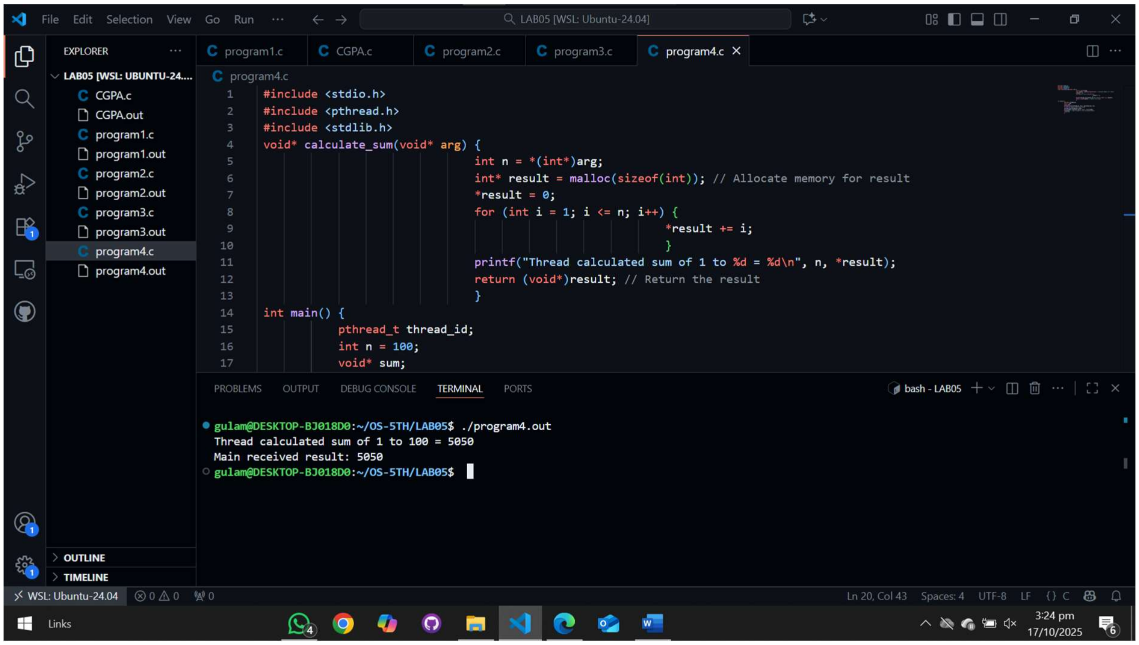
# Program 3: Name with  CGPA



```c
typedef struct {
                int id;
                float cgpa;
                char* message;
                } ThreadData;
void* printData(void* arg) {
                ThreadData* data = (ThreadData*)arg;
                printf("Thread %d with CGPA %f says: %s\n", data->id,data->cgpa, data->message);
                return NULL;
                            }
int main() {
        pthread_t t1, t2;
        ThreadData data1 = {1159,3.55, "My name is Gulam Rasool"};
        pthread_create(&t1, NULL, printData, &data1);
        pthread_join(t1, NULL);
        printf("All threads done.\n");
        return 0;
        }
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program3.out
Thread 1159 with CGPA 3.550000 says: My name is Gulam Rasool
All threads done.
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

# Program 4: Thread Return Values

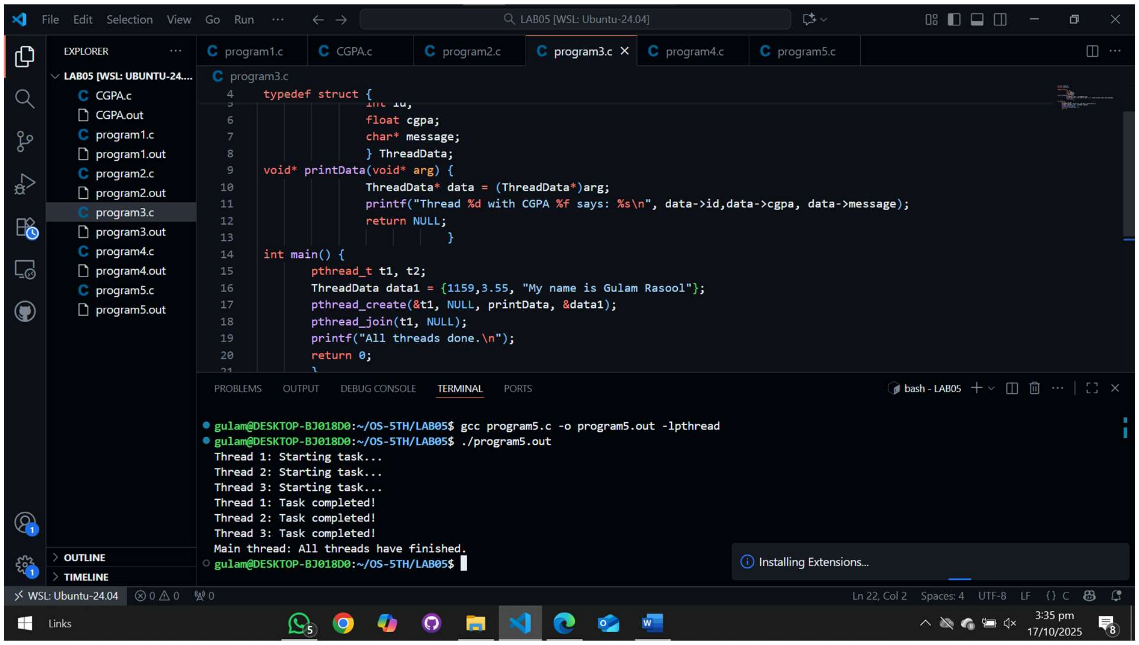


```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
                                int n = *(int*)arg;
                                int* result = malloc(sizeof(int)); // Allocate memory for result
                                *result = 0;
                                for (int i = 1; i <= n; i++) {
                                                            *result += i;
                                                            }
                                printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
                                return (void*)result; // Return the result
                                }
int main() {
           pthread_t thread_id;
           int n = 100;
           void* sum;
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program4.out
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ 
```

# Basic Multithreading

## Program 1: Creating and Running Multiple Threads

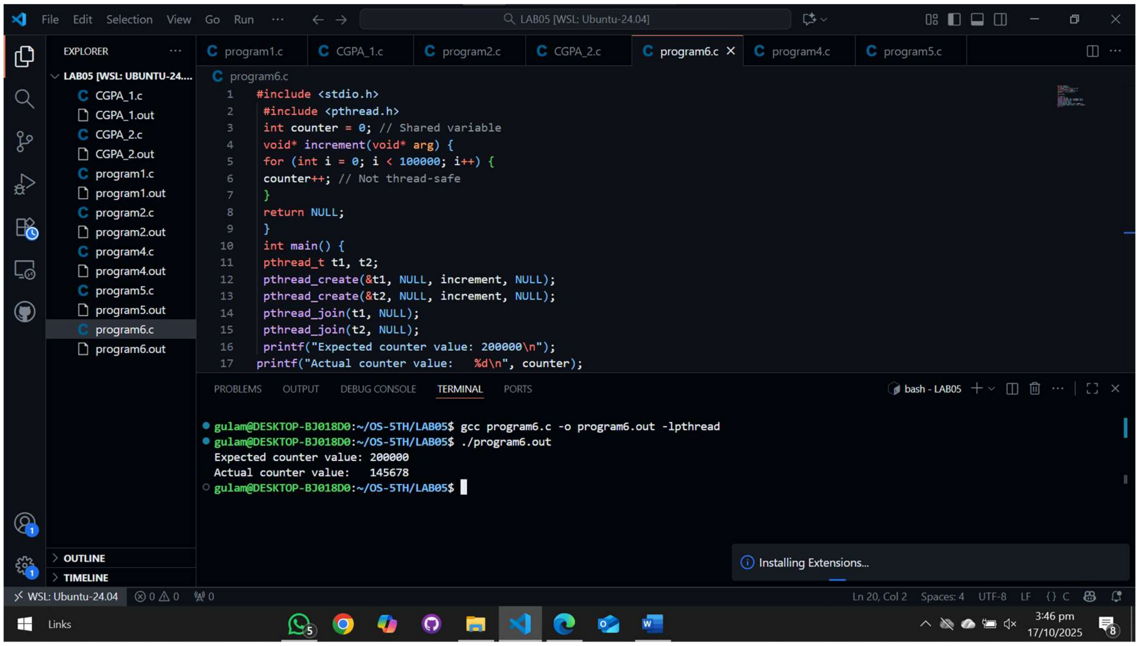


program3.c

```c
typedef struct {
               int id;
               float cgpa;
               char* message;
               } ThreadData;
void* printData(void* arg) {
               ThreadData* data = (ThreadData*)arg;
               printf("Thread %d with CGPA %f says: %s\n", data->id,data->cgpa, data->message);
               return NULL;
                           }
int main() {
       pthread_t t1, t2;
       ThreadData data1 = {1159,3.55, "My name is Gulam Rasool"};
       pthread_create(&t1, NULL, printData, &data1);
       pthread_join(t1, NULL);
       printf("All threads done.\n");
       return 0;
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program5.c -o program5.out -lpthread
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program5.out
Thread 1: Starting task...
Thread 2: Starting task...
Thread 3: Starting task...
Thread 1: Task completed!
Thread 2: Task completed!
Thread 3: Task completed!
Main thread: All threads have finished.
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

## Program 2: Demonstrating a Race Condition



program6.c

```c
#include <stdio.h>
 #include <pthread.h>
 int counter = 0; // Shared variable
 void* increment(void* arg) {
 for (int i = 0; i < 100000; i++) {
 counter++; // Not thread-safe
 }
 return NULL;
 }
 int main() {
 pthread_t t1, t2;
 pthread_create(&t1, NULL, increment, NULL);
 pthread_create(&t2, NULL, increment, NULL);
 pthread_join(t1, NULL);
 pthread_join(t2, NULL);
 printf("Expected counter value: 200000\n");
printf("Actual counter value:   %d\n", counter);
```

```
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program6.c -o program6.out -lpthread
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program6.out
Expected counter value: 200000
Actual counter value:   145678
gulam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```