

National Textile University

Department of Computer Science

Name

Gulam Rasool

Reg No:

23-NTU-CS-FL-1159

Semester:

5th

Lab:

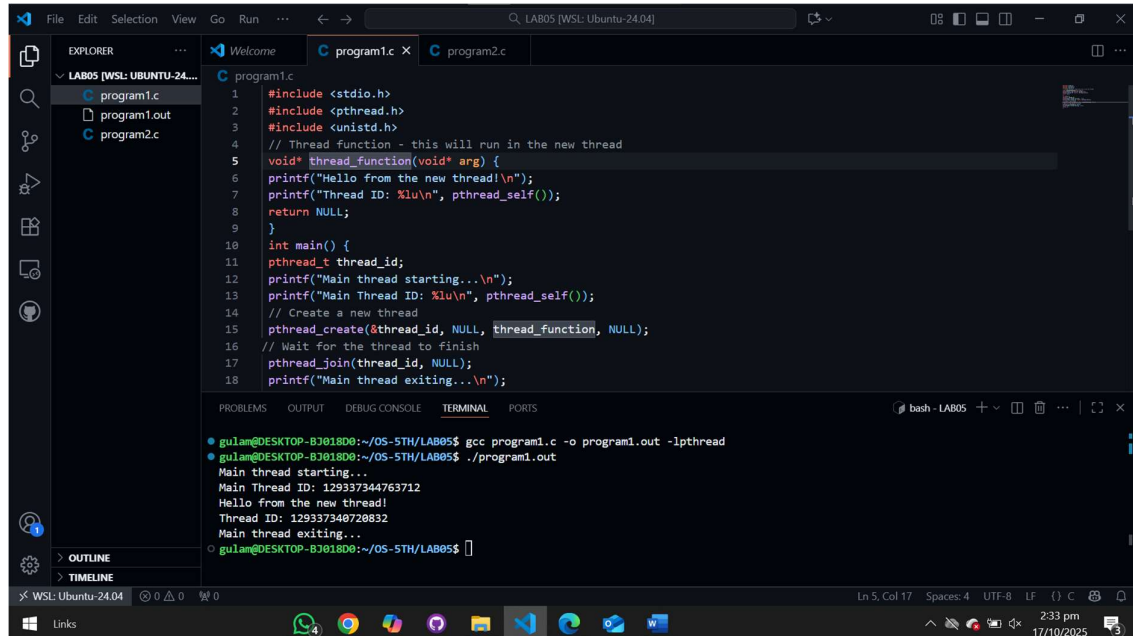
05-Class_Task

Submitted to:

Sir Nasir Mehmood

Lab-05(Class_Task)

Program 1: Creating a Simple Thread

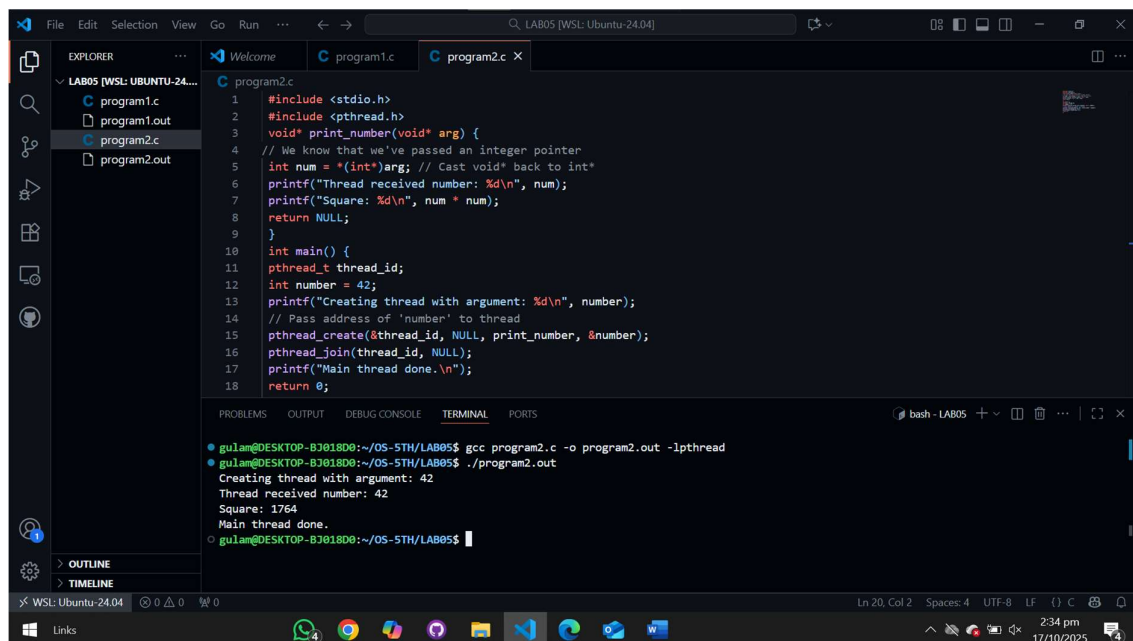


```
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <unistd.h>
4 // Thread function - this will run in the new thread
5 void* thread_function(void* arg) {
6     printf("Hello from the new thread!\n");
7     printf("Thread ID: %lu\n", pthread_self());
8     return NULL;
9 }
10 int main() {
11     pthread_t thread_id;
12     printf("Main thread starting...\n");
13     printf("Main Thread ID: %lu\n", pthread_self());
14     // Create a new thread
15     pthread_create(&thread_id, NULL, thread_function, NULL);
16     // Wait for the thread to finish
17     pthread_join(thread_id, NULL);
18     printf("Main thread exiting...\n");
19 }
```

Terminal Output:

```
gulam@DESKTOP-B3018D0:~/OS-5TH/LAB05$ gcc program1.c -o program1.out -lpthread
gulam@DESKTOP-B3018D0:~/OS-5TH/LAB05$ ./program1.out
Main thread starting...
Main Thread ID: 129337344763712
Hello from the new thread!
Thread ID: 129337340728832
Main thread exiting...
```

Program 2: Passing Arguments to Threads

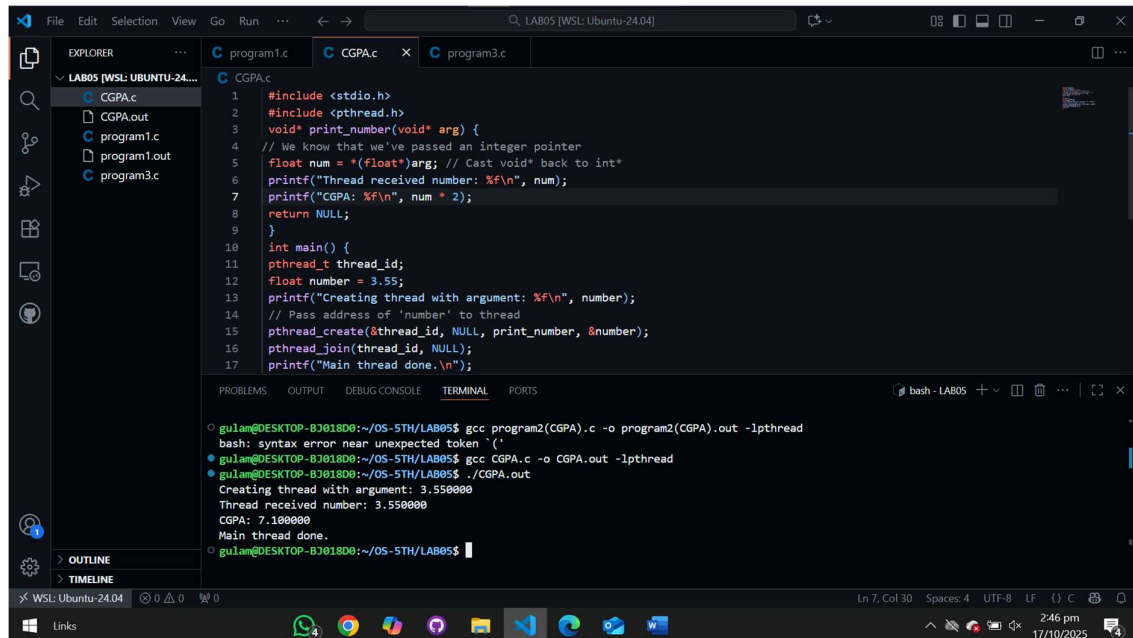


```
1 #include <stdio.h>
2 #include <pthread.h>
3 void* print_number(void* arg) {
4     // We know that we've passed an integer pointer
5     int num = *(int*)arg; // Cast void* back to int*
6     printf("Thread received number: %d\n", num);
7     printf("Square: %d\n", num * num);
8     return NULL;
9 }
10 int main() {
11     pthread_t thread_id;
12     int number = 42;
13     printf("Creating thread with argument: %d\n", number);
14     // Pass address of 'number' to thread
15     pthread_create(&thread_id, NULL, print_number, &number);
16     pthread_join(thread_id, NULL);
17     printf("Main thread done.\n");
18     return 0;
19 }
```

Terminal Output:

```
gulam@DESKTOP-B3018D0:~/OS-5TH/LAB05$ gcc program2.c -o program2.out -lpthread
gulam@DESKTOP-B3018D0:~/OS-5TH/LAB05$ ./program2.out
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
```

Program 2: Double CGPA

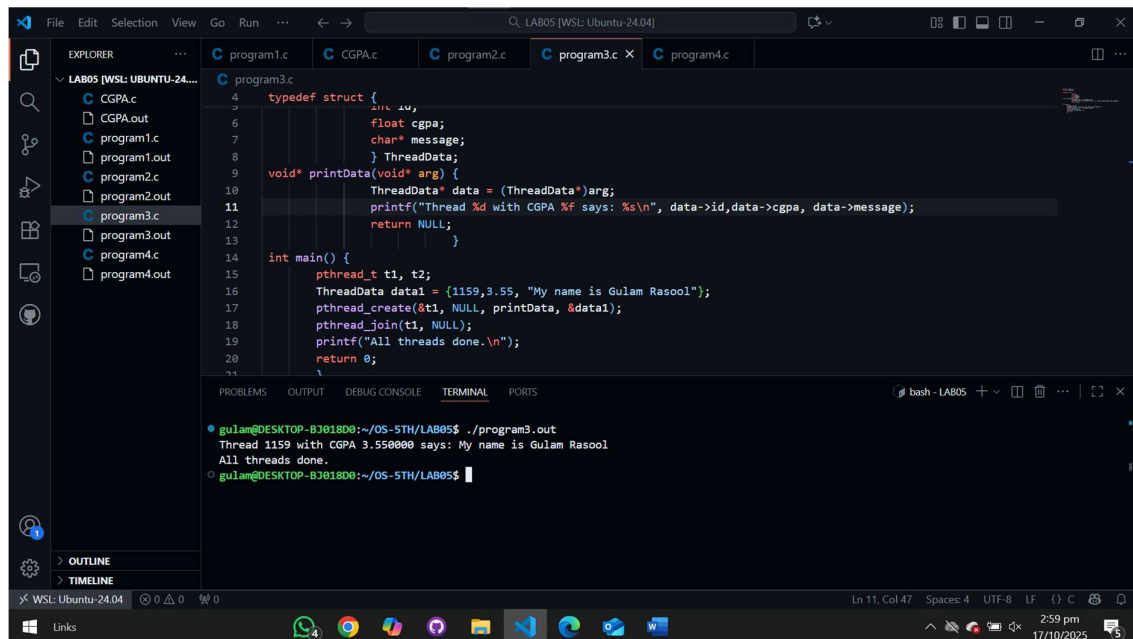


```
1 #include <stdio.h>
2 #include <pthread.h>
3 void* print_number(void* arg) {
4     // We know that we've passed an integer pointer
5     float num = *(float*)arg; // Cast void* back to int*
6     printf("Thread received number: %f\n", num);
7     printf("CGPA: %f\n", num * 2);
8     return NULL;
9 }
10 int main() {
11     pthread_t thread_id;
12     float number = 3.55;
13     printf("Creating thread with argument: %f\n", number);
14     // Pass address of 'number' to thread
15     pthread_create(&thread_id, NULL, print_number, &number);
16     pthread_join(thread_id, NULL);
17     printf("Main thread done.\n");
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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

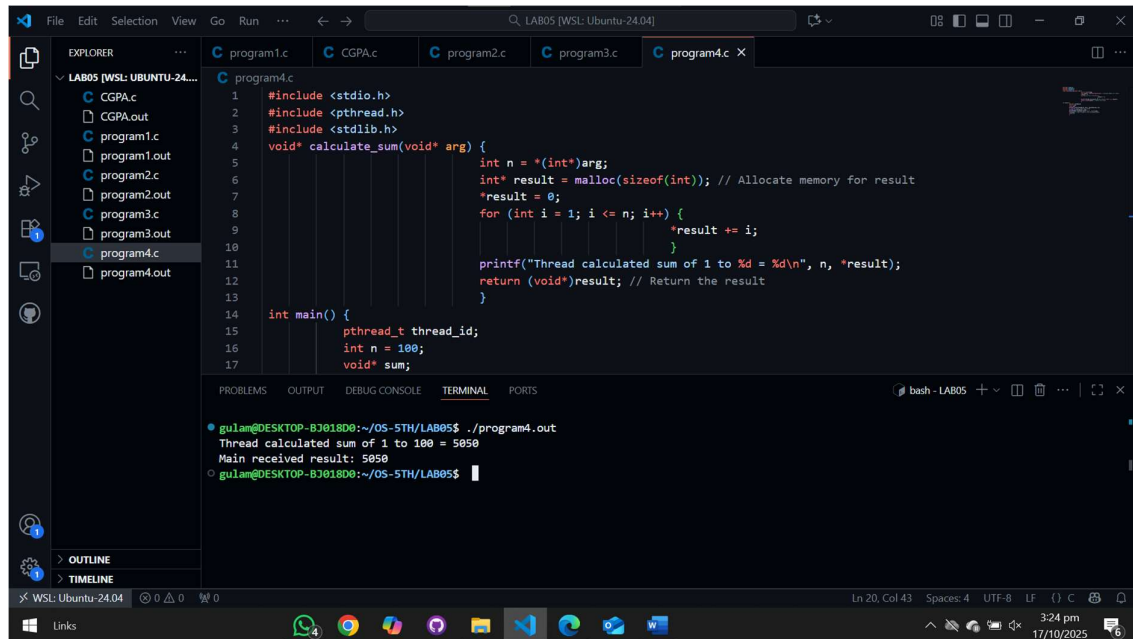


```
1 typedef struct {
2     int id;
3     float cgpa;
4     char* message;
5 } ThreadData;
6 void* printData(void* arg) {
7     ThreadData* data = (ThreadData*)arg;
8     printf("Thread %d with CGPA %f says: %s\n", data->id, data->cgpa, data->message);
9     return NULL;
10 }
11 int main() {
12     pthread_t t1, t2;
13     ThreadData data1 = {1159, 3.55, "My name is Gulam Rasool"};
14     pthread_create(&t1, NULL, printData, &data1);
15     pthread_join(t1, NULL);
16     printf("All threads done.\n");
17     return 0;
18 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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



The screenshot displays the Visual Studio Code interface with a C program named `program4.c` open. The program calculates the sum of integers from 1 to 100 using a thread. The code is as follows:

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <stdlib.h>
4  void* calculate_sum(void* arg) {
5      int n = *(int*)arg;
6      int* result = malloc(sizeof(int)); // Allocate memory for result
7      *result = 0;
8      for (int i = 1; i <= n; i++) {
9          *result += i;
10     }
11     printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
12     return (void*)result; // Return the result
13 }
14
15 int main() {
16     pthread_t thread_id;
17     int n = 100;
18     void* sum;
```

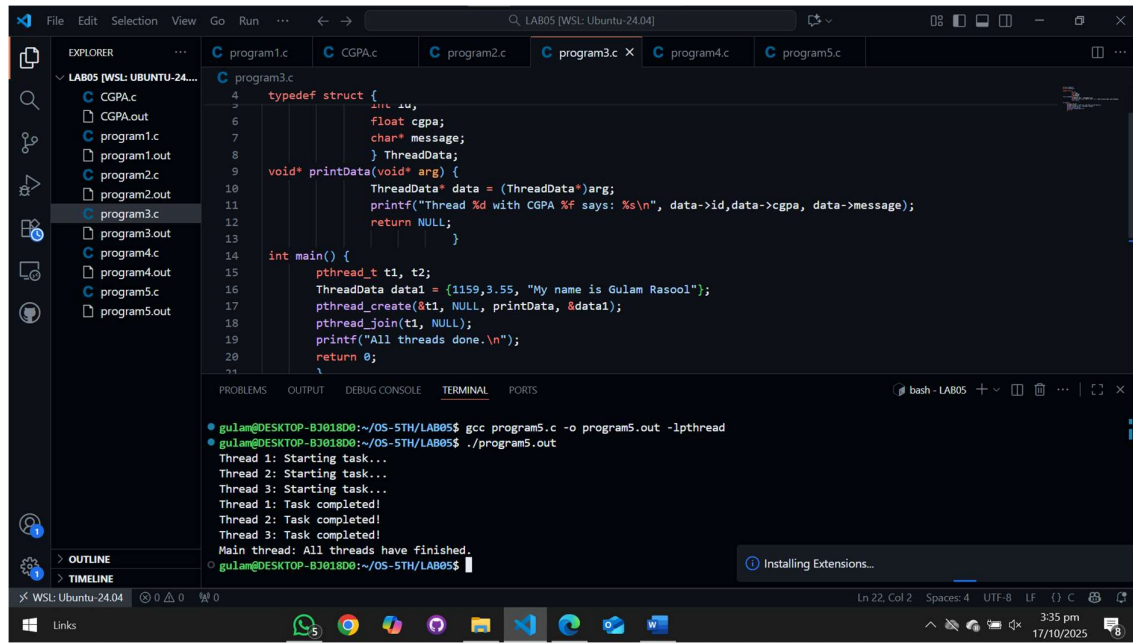
The terminal output shows the execution of the program:

```
gulam@DESKTOP-B3018D0:~/OS-STH/LAB05$ ./program4.out
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
gulam@DESKTOP-B3018D0:~/OS-STH/LAB05$
```

The status bar at the bottom indicates the file is at line 20, column 43, with 4 spaces, UTF-8 encoding, and LF line endings.

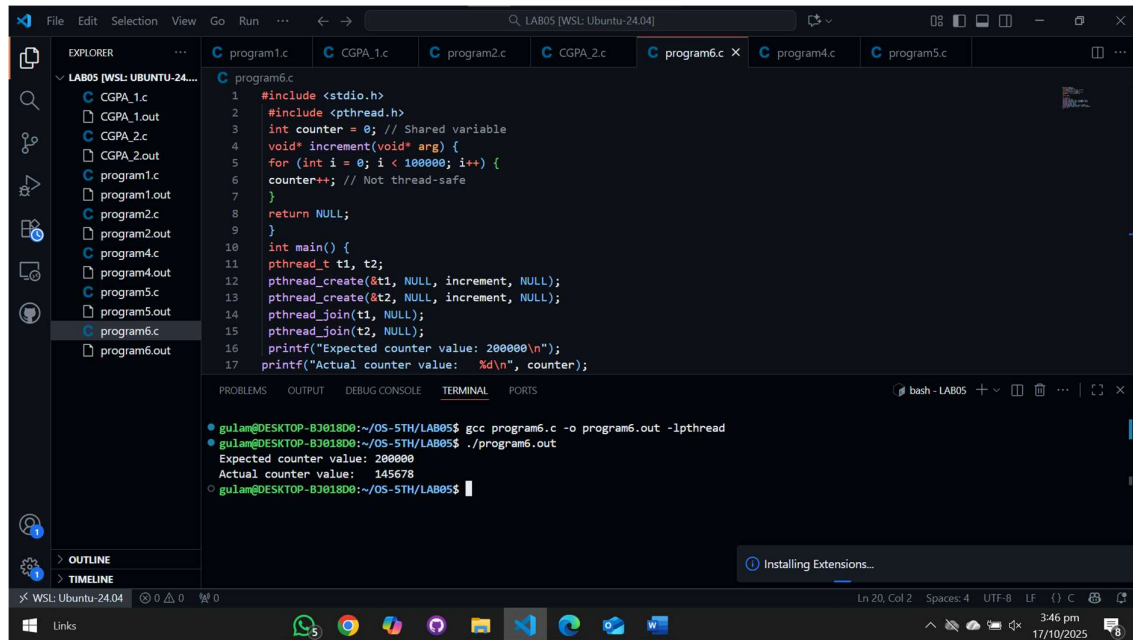
Basic Multithreading

Program 1: Creating and Running Multiple Threads



```
File Edit Selection View Go Run ... LAB05 [WSL: Ubuntu-24.04]
EXPLORER
  LAB05 [WSL: UBUNTU-24...
    CGPA.c
    CGPA.out
    program1.c
    program1.out
    program2.c
    program2.out
    program3.c
    program3.out
    program4.c
    program4.out
    program5.c
    program5.out
  program3.c
    4 typedef struct {
    5     int id;
    6     float cgpa;
    7     char* message;
    8 } ThreadData;
    9 void* printData(void* arg) {
    10     ThreadData* data = (ThreadData*)arg;
    11     printf("Thread %d with CGPA %f says: %s\n", data->id, data->cgpa, data->message);
    12     return NULL;
    13 }
    14 int main() {
    15     pthread_t t1, t2;
    16     ThreadData data1 = {1159, 3.55, "My name is Gulam Rasool"};
    17     pthread_create(&t1, NULL, printData, &data1);
    18     pthread_join(t1, NULL);
    19     printf("All threads done.\n");
    20     return 0;
    21 }
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
  bash - LAB05
  guilam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program5.c -o program5.out -lpthread
  guilam@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.
  guilam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```

Program 2: Demonstrating a Race Condition



```
File Edit Selection View Go Run ... LAB05 [WSL: Ubuntu-24.04]
EXPLORER
  LAB05 [WSL: UBUNTU-24...
    CGPA_1.c
    CGPA_1.out
    CGPA_2.c
    CGPA_2.out
    program1.c
    program1.out
    program2.c
    program2.out
    program4.c
    program4.out
    program5.c
    program5.out
    program6.c
    program6.out
  program6.c
    1 #include <stdio.h>
    2 #include <pthread.h>
    3 int counter = 0; // Shared variable
    4 void* increment(void* arg) {
    5     for (int i = 0; i < 100000; i++) {
    6         counter++; // Not thread-safe
    7     }
    8     return NULL;
    9 }
    10 int main() {
    11     pthread_t t1, t2;
    12     pthread_create(&t1, NULL, increment, NULL);
    13     pthread_create(&t2, NULL, increment, NULL);
    14     pthread_join(t1, NULL);
    15     pthread_join(t2, NULL);
    16     printf("Expected counter value: 200000\n");
    17     printf("Actual counter value: %d\n", counter);
    18 }
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
  bash - LAB05
  guilam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ gcc program6.c -o program6.out -lpthread
  guilam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$ ./program6.out
  Expected counter value: 200000
  Actual counter value: 145678
  guilam@DESKTOP-BJ018D0:~/OS-5TH/LAB05$
```