# Operating Systems(CSL3030)

## InLab Assignment 6 - Report

Soumen Kumar
Roll No-B22ES006

## 1 Introduction

Multithreading is a method where multiple threads are created to perform different tasks in a program. In C, the POSIX library 'pthread.h' provides the necessary functions for creating and managing threads. This report explains a simple C program that uses multithreading to compute the sum of two numbers.

## 2 Code Explanation

The code below demonstrates how a new thread is created to compute the sum of two user-input integers. The main thread collects the input from the user and passes it to the newly created thread, which performs the calculation.

```c
#include <stdio.h>
#include <pthread.h>

typedef struct {
    int num1;
    int num2;
    int result;
} SumArgs;

// Thread function to compute the sum
void* compute_sum(void* arg) {
    SumArgs* args = (SumArgs*)arg;
    args->result = args->num1 + args->num2;
    pthread_exit(NULL);
}

int main() {
    pthread_t thread;
    SumArgs args;
    int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);//taking input from the user and storing it in a and b(main thread)
    args.num1 =a;
    args.num2 =b;
    pthread_create(&thread, NULL, compute_sum, &args);//creating a thread(child) that will compute the sum
     of the numbers and passing the argument from main thread
    pthread_join(thread, NULL);//this ensures the parent process waits for the thread to finish
    printf("The sum is: %d\n", args.result);
    return 0;
}
```

## 3 Explanation of Key Functions

- `pthread_create()`: This function is used to create a new thread. It takes four arguments: a pointer to the thread identifier, thread attributes (which is set to `NULL` for default attributes), the function to be executed in the thread, and the arguments to be passed to that function.
- `pthread_join()`: This function makes the calling thread (main thread) wait for the specified thread to finish execution.
- `pthread_exit()`: This is called by the thread function to terminate the thread and exit.

## 4 Program Workflow

The program follows the steps below:

1. The main thread prompts the user to enter two numbers and stores them in variables.

2. The input values are set in a structure `SumArgs`, which holds the two numbers to be added.
3. A new thread is created using `pthread_create()`. The thread executes the function `compute_sum()`.
4. Inside the thread function `compute_sum()`, the sum of the two numbers is computed and stored in the structure.
5. The main thread waits for the newly created thread to complete using `pthread_join()`.
6. After the thread completes, the main thread prints the computed sum.

# 5 Screenshots

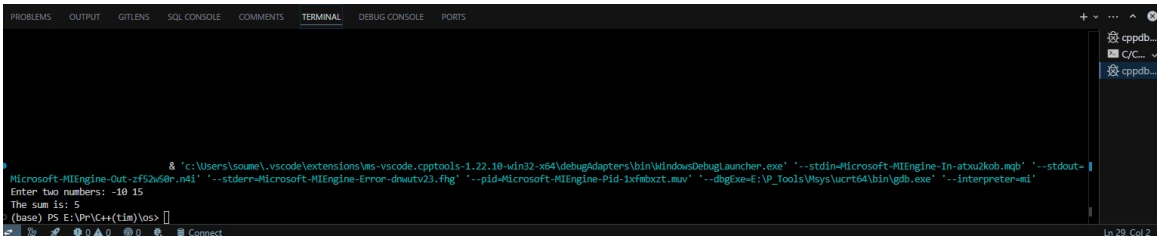


Figure 1: Test Case 1


Figure 2: Test Case 2

# 6 Conclusion

This program demonstrates the use of multithreading in C by creating a separate thread to perform a task (computing the sum of two integers) while the main thread waits for the result. Multithreading is a useful technique for parallel execution in modern applications, allowing tasks to be distributed across multiple threads for better performance and responsiveness.