Lab 3: POSIX Thread Basics

Objective. To introduce the fundamental operations of POSIX threads in C under Linux, focusing on thread creation with `pthread_create`, termination with `pthread_exit`, and simple synchronization using `pthread_join`. The lab covers managing the lifecycle of threads in a concurrent environment.

Exercise 1

Write a program that creates a thread which prints the message ”HELLO”.
- Modify the program so that the thread prints a message provided by the user, handling both cases with and without argument passing to the thread function.

Exercise 2

Write a program that calculates the sum and product of two integers provided by the user. The program should create two threads: one to calculate and print the sum, and the other to calculate and print the product.
- Modify the program so that the main thread prints the results instead of the threads.
- Modify the program so that the two numbers are passed as arguments to the threads, and the main thread prints the results.

Exercise 3

Write a program that reads an integer provided by the user and prints its double in the following scenarios:

1. The main program creates two threads. The first thread reads the number and calculates its double, while the second thread prints the result.
2. The main program creates one thread that takes the number as an argument, calculates its double, and prints the result.
3. The main program creates one thread that takes the number as an argument, calculates its double, and returns the result to the main program, which then prints it.

Exercise 4

Explain what the following program does.

```c
#include <stdio.h>
#include <pthread.h>
#define SIZE 6
int tab[SIZE] = {1, 2, 3, 4, 5, 6};
int sum = 0;
void *calculate_sum(void *arg);
int main() {
    pthread_t th1, th2;
    int start1 = 0;
    int start2 = SIZE/2;
    pthread_create(&th1, NULL, calculate_sum, (void *)&start1);
    pthread_create(&th2, NULL, calculate_sum, (void *)&start2);
    pthread_join(th1, NULL);
    pthread_join(th2, NULL);
    printf("The sum is: %d\n", sum);
    return 0;}
void *calculate_sum(void *arg) {
    int start = *((int *)arg);
    int end = start + SIZE/2;
    for (int i = start; i < end; i++) {
        sum += tab[i]; }
    return NULL;}
```