



National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Naseer

Submitted by:

Ayema

Reg number:

23-NTU-CS-1142

Lab no. :

04

Semester: 5Th

Program 1: Creating a Simple Thread:

```
#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

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());

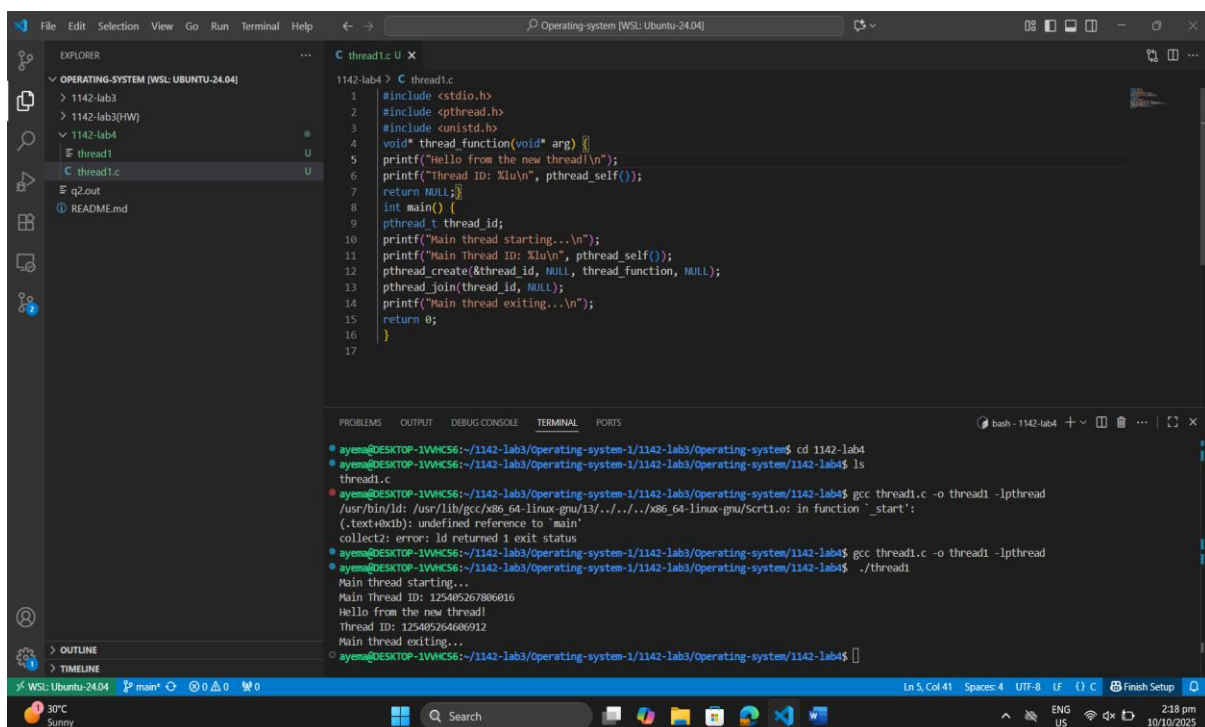
pthread_create(&thread_id, NULL, thread_function, NULL);

pthread_join(thread_id, NULL);

printf("Main thread exiting...\n");

return 0;

}
```



The screenshot shows a WSL terminal window with the following content:

```
1142-lab4 > C thread1.c
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <unistd.h>
4 void* thread_function(void* arg) {
5 printf("Hello from the new thread!\n");
6 printf("Thread ID: %lu\n", pthread_self());
7 return NULL;}
8 int main() {
9 pthread_t thread_id;
10 printf("Main thread starting...\n");
11 printf("Main Thread ID: %lu\n", pthread_self());
12 pthread_create(&thread_id, NULL, thread_function, NULL);
13 pthread_join(thread_id, NULL);
14 printf("Main thread exiting...\n");
15 return 0;
16 }
17
```

The terminal output shows the following commands and results:

```
ayema@DESKTOP-1VHKS6:~/1142-lab3/Operating-system-1/1142-lab3/Operating-system$ cd 1142-lab4
ayema@DESKTOP-1VHKS6:~/1142-lab3/Operating-system-1/1142-lab3/Operating-system/1142-lab4$ ls
thread1.c
ayema@DESKTOP-1VHKS6:~/1142-lab3/Operating-system-1/1142-lab3/Operating-system/1142-lab4$ gcc thread1.c -o thread1 -lpthread
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/13/../../../../x86_64-linux-gnu/Scrt1.o: in function `_start':
(.text+0x1b): undefined reference to `main'
collect2: error: ld returned 1 exit status
ayema@DESKTOP-1VHKS6:~/1142-lab3/Operating-system-1/1142-lab3/Operating-system/1142-lab4$ gcc thread1.c -o thread1 -lpthread
ayema@DESKTOP-1VHKS6:~/1142-lab3/Operating-system-1/1142-lab3/Operating-system/1142-lab4$ ./thread1
Main thread starting...
Main Thread ID: 125405267806016
Hello from the new thread!
Thread ID: 125405264060912
Main thread exiting...
```

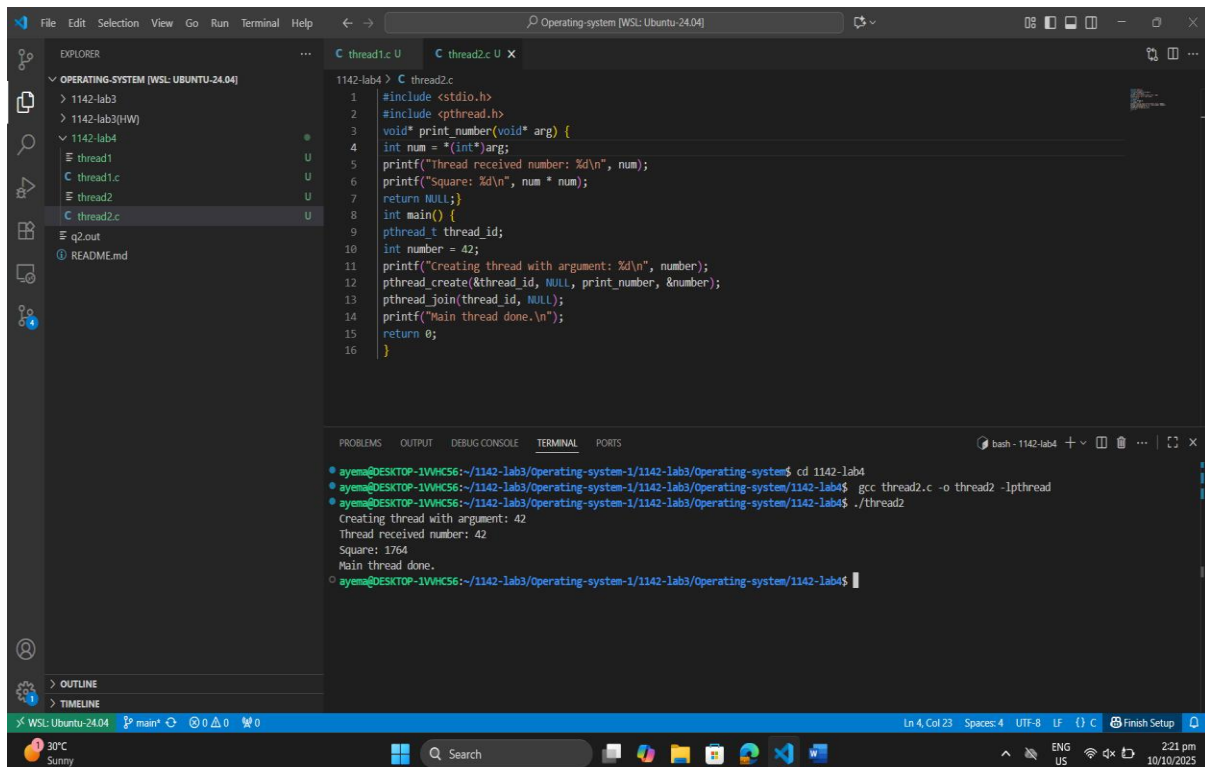
Program 2: Passing Arguments to Threads:

```
#include <stdio.h>

#include <pthread.h>

void* print_number(void* arg) {
    int num = *(int*)arg;
    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);
    pthread_create(&thread_id, NULL, print_number, &number);
    pthread_join(thread_id, NULL);
    printf("Main thread done.\n");
    return 0; }
```



The screenshot displays the Visual Studio Code interface with a C program for passing arguments to threads. The code is written in `thread2.c` and is shown in the editor window. The program includes `<stdio.h>` and `<pthread.h>`, defines a `print_number` function that takes a `void*` argument, casts it to `int*`, prints the number and its square, and returns `NULL`. The `main` function creates a thread with the `print_number` function and the address of a variable `number` (set to 42) as arguments, joins the thread, and prints "Main thread done."

The terminal output shows the execution of the program:

```
ayem@DESKTOP-1VHKS6:~/1142-lab3/operating-system-1/1142-lab3/operating-system$ cd 1142-lab4
ayem@DESKTOP-1VHKS6:~/1142-lab3/operating-system-1/1142-lab3/operating-system/1142-lab4$ gcc thread2.c -o thread2 -lpthread
ayem@DESKTOP-1VHKS6:~/1142-lab3/operating-system-1/1142-lab3/operating-system/1142-lab4$ ./thread2
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
```

Program 3: Passing Multiple Data

```
#include <stdio.h>

#include <pthread.h>

typedef struct {
    int id;
    char* message;
} ThreadData;

void* printData(void* arg) {
    ThreadData* data = (ThreadData*)arg;
    printf("Thread %d says: %s\n", data->id, data->message);
    return NULL;}

int main() {
    pthread_t t1, t2;
    ThreadData data1 = {1, "Hello"};
    ThreadData data2 = {2, "World"};
    pthread_create(&t1, NULL, printData, &data1);
    pthread_create(&t2, NULL, printData, &data2);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    printf("All threads done.\n");
    return 0;}
```

