



National Textile University
Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir

Submitted by:

Ahmad Fawad

Reg number:

1129

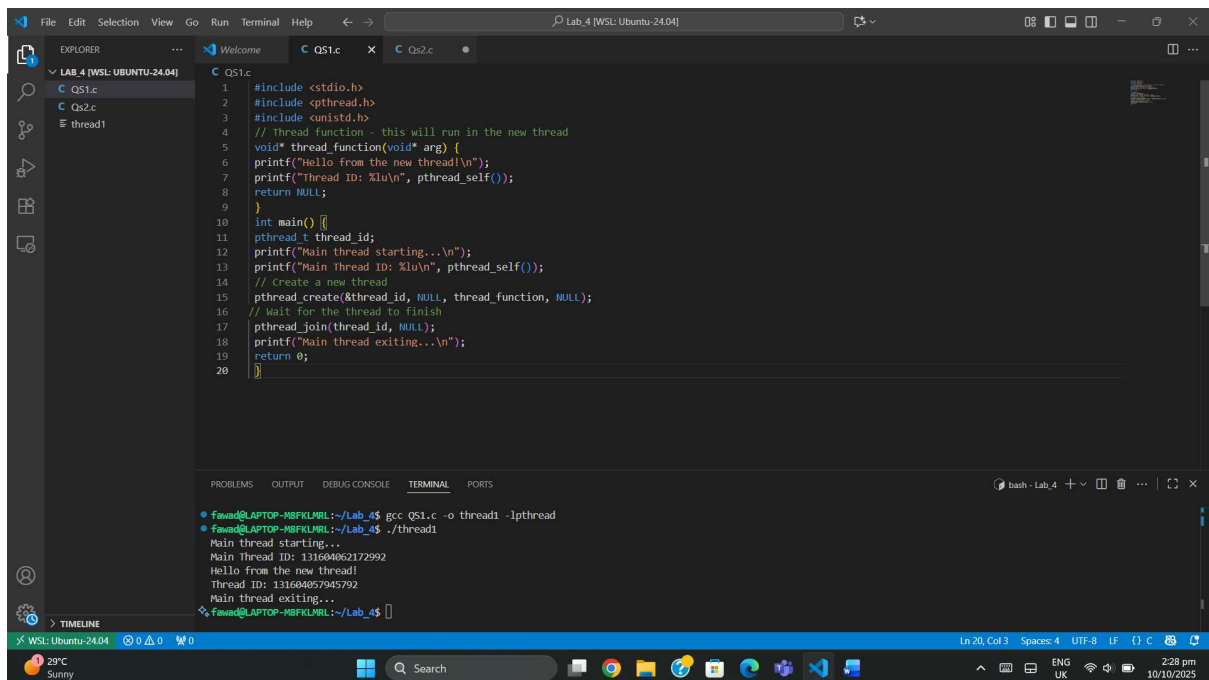
Lab no. :

04

Semester:

5th

Question 1:

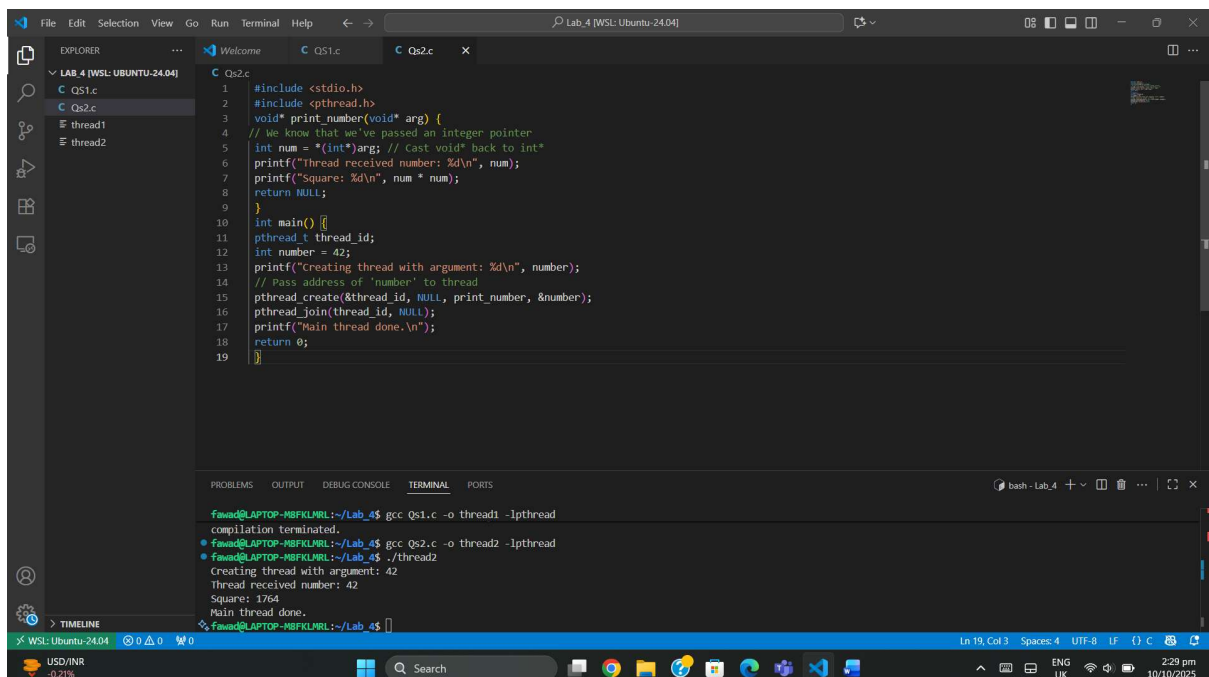


The screenshot shows a Visual Studio Code editor with a C file named `Qs1.c`. The code implements a thread function `thread_function` that prints a message and its ID, and a `main` function that creates and joins this thread. The terminal output shows the successful execution of the program, displaying the thread's ID and the main thread's exit message.

```
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     return 0;
20 }
```

```
faewad@LAPTOP-MBFKLMRL:~/Lab_4$ gcc Qs1.c -o thread1 -lpthread
faewad@LAPTOP-MBFKLMRL:~/Lab_4$ ./thread1
Main thread starting...
Main thread ID: 131694862172992
Hello from the new thread!
Thread ID: 131694862172992
Main thread exiting...
```

Question 2:



The screenshot shows a Visual Studio Code editor with a C file named `Qs2.c`. The code implements a thread function `print_number` that takes an integer pointer as an argument, prints the number and its square, and a `main` function that creates and joins this thread with an argument. The terminal output shows the successful execution of the program, displaying the thread's argument and the main thread's exit message.

```
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 }
```

```
faewad@LAPTOP-MBFKLMRL:~/Lab_4$ gcc Qs1.c -o thread1 -lpthread
faewad@LAPTOP-MBFKLMRL:~/Lab_4$ gcc Qs2.c -o thread2 -lpthread
faewad@LAPTOP-MBFKLMRL:~/Lab_4$ ./thread2
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
```