



National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir

Submitted by:

Esha Mubashir Khan

Reg number:

23-NTU-CS-1151

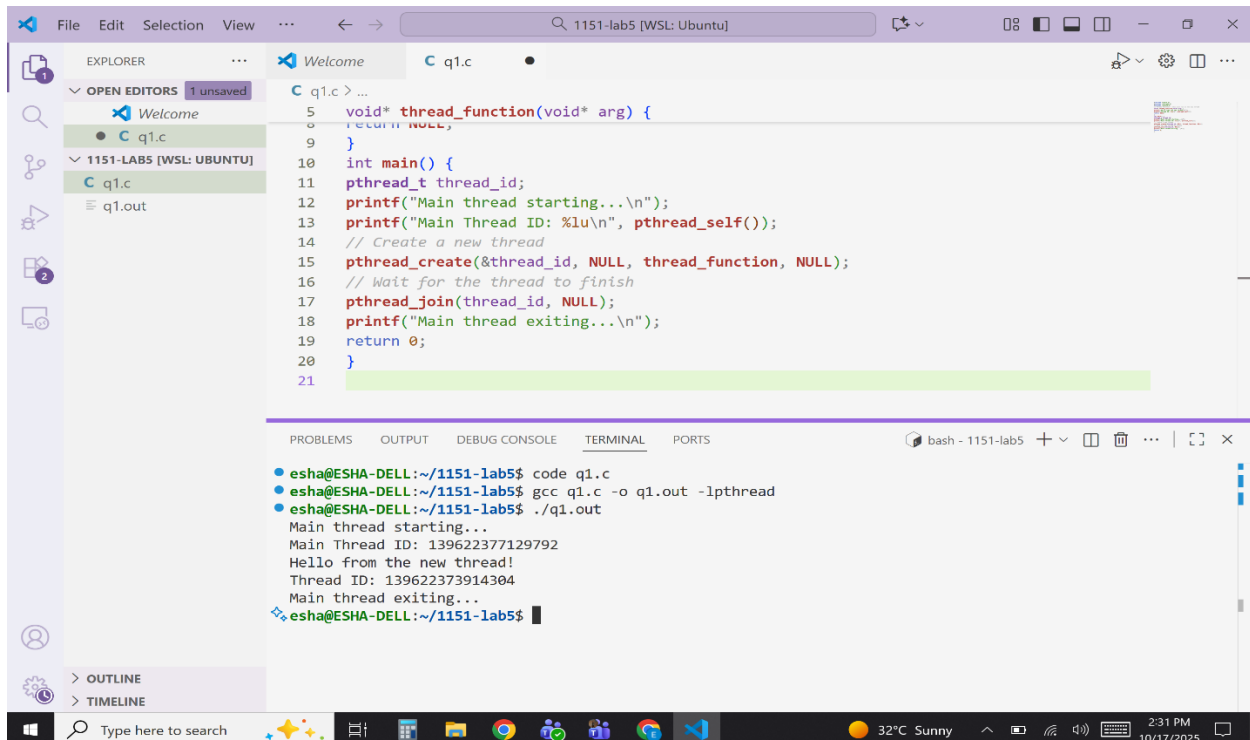
Lab no: 5

Semester: 5th

3. C Programs with Threads

Program 1: Creating a Simple Thread

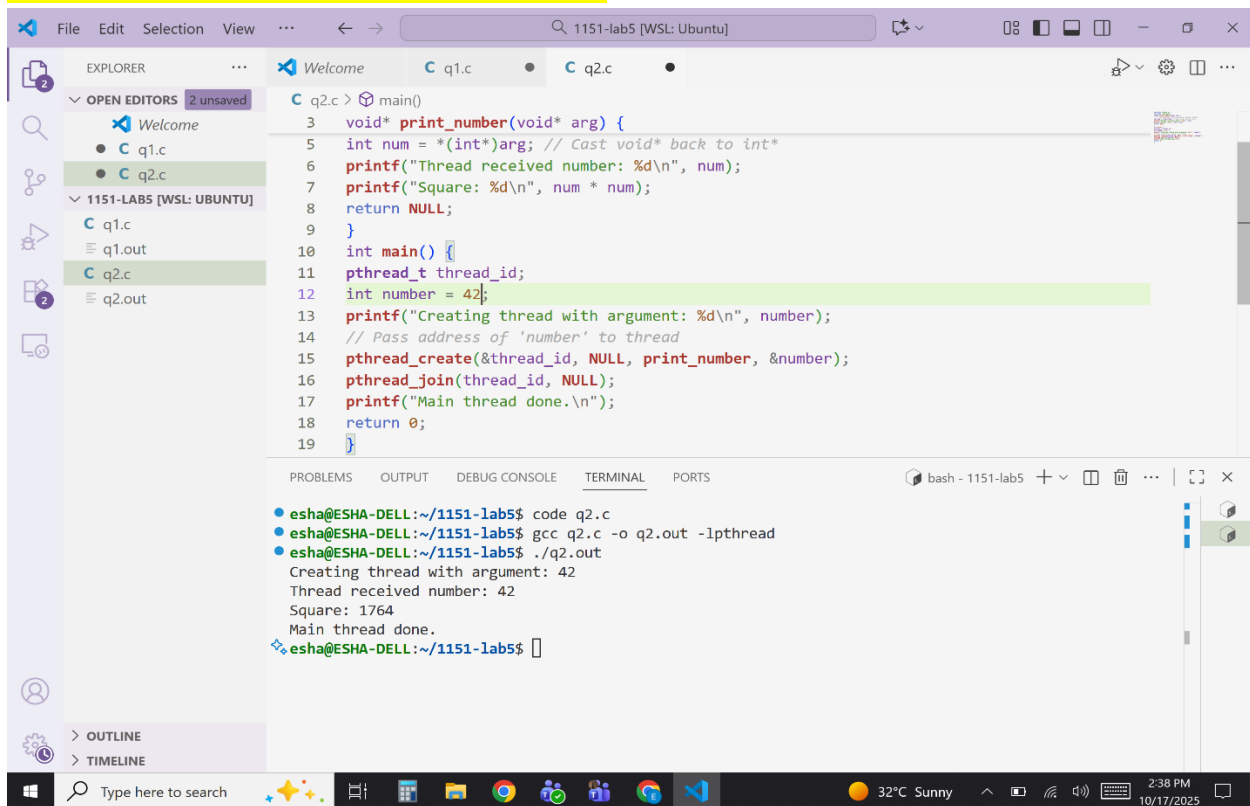
```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <unistd.h>
4  void* thread_function(void* arg) {
5      printf("Hello from the new thread!\n");
6      printf("Thread ID: %lu\n", pthread_self());
7      return NULL;
8  }
9  int main() {
10     pthread_t thread_id;
11     printf("Main thread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



The screenshot displays a Windows Subsystem for Linux (WSL) environment. The Explorer view on the left shows the file structure with 'q1.c' and 'q1.out' in the '1151-LAB5 [WSL: UBUNTU]' directory. The main editor window shows the C code for 'q1.c'. The terminal at the bottom shows the execution of the program:

```
esha@ESHA-DELL:~/1151-lab5$ code q1.c
esha@ESHA-DELL:~/1151-lab5$ gcc q1.c -o q1.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q1.out
Main thread starting...
Main Thread ID: 139622377129792
Hello from the new thread!
Thread ID: 139622373914304
Main thread exiting...
esha@ESHA-DELL:~/1151-lab5$
```

Program 2: Passing Arguments to Threads



```
File Edit Selection View ... 1151-lab5 [WSL: Ubuntu]
EXPLORER
  OPEN EDITORS 2 unsaved
    Welcome
    C q1.c
    C q2.c
  1151-LABS [WSL: UBUNTU]
    C q1.c
    C q2.c
    q1.out
    q2.out

C q2.c > main()
3 void* print_number(void* arg) {
4   int num = *(int*)arg; // Cast void* back to int*
5   printf("Thread received number: %d\n", num);
6   printf("Square: %d\n", num * num);
7   return NULL;
8 }
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 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - 1151-lab5
esha@ESHA-DELL:~/1151-lab5$ code q2.c
esha@ESHA-DELL:~/1151-lab5$ gcc q2.c -o q2.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
esha@ESHA-DELL:~/1151-lab5$
```

PRINT CGPA:

```
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
    float num = *(float*)arg;
    printf("Thread received number: %.2f\n", num);
    printf("Double: %.2f\n", num + num);
    return NULL;
}
int main() {
    pthread_t thread_id;
    float number = 3.1;
    printf("Creating thread with argument: %.2f\n", number);
    pthread_create(&thread_id, NULL, print_number, &number);
    pthread_join(thread_id, NULL);
    printf("Main thread done.\n");
    return 0;
}
```

File Edit Selection View ... 1151-lab5 [WSL: Ubuntu]

EXPLORER

OPEN EDITORS 1 unsaved

- Welcome
- q1.c
- q2.c

1151-LABS [WSL: UBUNTU]

- q1.c
- q1.out
- q2.c
- q2.out

q2.c > main()

```
1 void* print_number(void* arg) {
2     // We know that we've passed an integer pointer
3     float num = *(float*)arg; // Cast void* back to float*
4     printf("Thread received number: %.2f\n", num);
5     printf("Double: %.2f\n", num + num);
6     return NULL;
7 }
8
9 int main() {
10     pthread_t thread_id;
11     float number = 3.1;
12     printf("Creating thread with argument: %.2f\n", number);
13     // Pass address of 'number' to thread
14     pthread_create(&thread_id, NULL, print_number, &number);
15     pthread_join(thread_id, NULL);
16     printf("Main thread done.\n");
17     return 0;
18 }
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - 1151-lab5

```
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
esha@ESHA-DELL:~/1151-lab5$ gcc q2.c -o q2.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q2.out
Creating thread with argument: 3.10
Thread received number: 3.10
Double: 6.20
Main thread done.
esha@ESHA-DELL:~/1151-lab5$
```

Type here to search 32°C Sunny 2:47 PM 10/17/2025

Program 3: Passing Multiple Data

File Edit Selection View ... 1151-lab5 [WSL: Ubuntu]

EXPLORER

OPEN EDITORS 2 unsaved

- Welcome
- q1.c
- q2.c
- q3.c

1151-LABS [WSL: UBUNTU]

- q1.c
- q1.out
- q2.c
- q2.out
- q3.c
- q3.out

q3.c > ...

```
7 void* printData(void* arg) {
8     ThreadData* data = (ThreadData*)arg;
9     printf("Thread %d says: %s\n", data->id, data->message);
10    return NULL;
11 }
12
13 int main() {
14     pthread_t t1, t2;
15     ThreadData data1 = {1, "Hello"};
16     ThreadData data2 = {2, "World"};
17     pthread_create(&t1, NULL, printData, &data1);
18     pthread_create(&t2, NULL, printData, &data2);
19     pthread_join(t1, NULL);
20     pthread_join(t2, NULL);
21     printf("All threads done.\n");
22     return 0;
23 }
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - 1151-lab5

```
esha@ESHA-DELL:~/1151-lab5$ code q3.c
esha@ESHA-DELL:~/1151-lab5$ gcc q3.c -o q3.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
Thread 1 says: Hello
Thread 2 says: World
All threads done.
esha@ESHA-DELL:~/1151-lab5$
```

Type here to search 32°C Sunny 2:49 PM 10/17/2025

PRINT NAME & CGPA:

The screenshot shows the Visual Studio Code interface with a C program in `q2.c` and a terminal window showing the execution of `q3.out`.

EXPLORER

- 1151-Lab5
 - q1.c
 - q1.out
 - q2.c
 - q2.out
 - q3.c
 - q3.out

q2.c

```
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 float*
6     printf("Thread received number: %.2f\n", num);
7     printf("Double: %.2f\n", num + num);
8     return NULL;
9 }
10 int main() {
11     pthread_t thread_id;
12     float number = 3.1;
13     printf("Creating thread with argument: %.2f\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 }
```

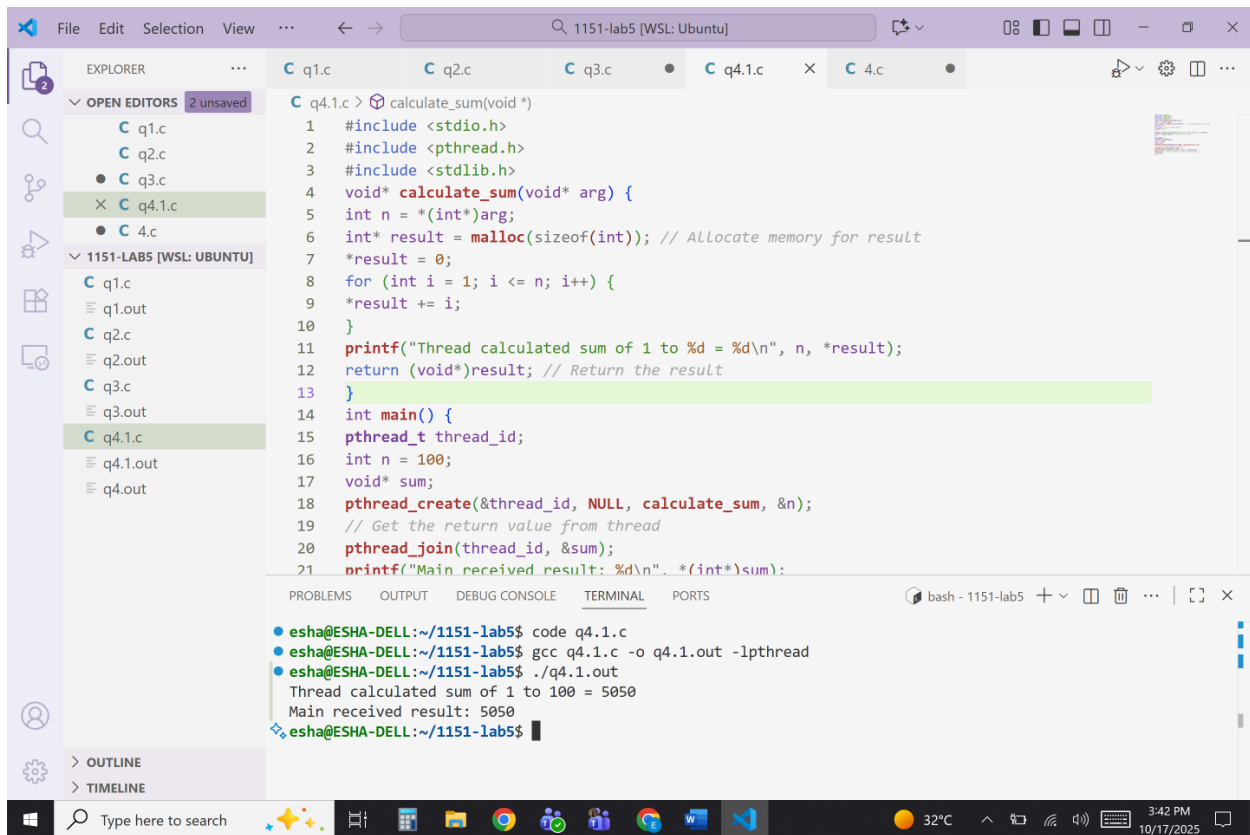
TERMINAL

```
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
All threads done.
esha@ESHA-DELL:~/1151-lab5$ code q3.c
esha@ESHA-DELL:~/1151-lab5$ gcc q3.c -o q3.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q3.out
Thread 1 says: ESHA
Thread 2 says: 3.00
All threads done.
esha@ESHA-DELL:~/1151-lab5$
```

Program 4.1: Thread Return Values



```
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 int main() {
15     pthread_t thread_id;
16     int n = 100;
17     void* sum;
18     pthread_create(&thread_id, NULL, calculate_sum, &n);
19     // Get the return value from thread
20     pthread_join(thread_id, &sum);
21     printf("Main received result: %d\n", *(int*)sum);
22     free(sum); // Don't forget to free allocated memory
23     return 0;
24 }
```



```
File Edit Selection View ... 1151-lab5 [WSL: Ubuntu]
EXPLORER
OPEN EDITORS 2 unsaved
q1.c
q2.c
q3.c
q4.1.c
4.c
1151-LABS [WSL: UBUNTU]
q1.c
q1.out
q2.c
q2.out
q3.c
q3.out
q4.1.c
q4.1.out
q4.out
OUTLINE
TIMELINE
q1.c
q2.c
q3.c
q4.1.c
4.c
q4.1.c
calculate_sum(void *)
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 int main() {
15     pthread_t thread_id;
16     int n = 100;
17     void* sum;
18     pthread_create(&thread_id, NULL, calculate_sum, &n);
19     // Get the return value from thread
20     pthread_join(thread_id, &sum);
21     printf("Main received result: %d\n", *(int*)sum);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - 1151-lab5
esha@ESHA-DELL:~/1151-lab5$ code q4.1.c
esha@ESHA-DELL:~/1151-lab5$ gcc q4.1.c -o q4.1.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q4.1.out
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
esha@ESHA-DELL:~/1151-lab5$
```

Program 1: Creating and Running Multiple Threads



```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <unistd.h>
4  void* worker(void* arg) {
5  int thread_num = *(int*)arg;
6  printf("Thread %d: Starting task...\n", thread_num);
7  sleep(1); // Simulate some work
8  printf("Thread %d: Task completed!\n", thread_num);
9  return NULL;
10 }
11 int main() {
12 pthread_t threads[3];
13 int thread_ids[3];
14 for (int i = 0; i < 3; i++) {
15 thread_ids[i] = i + 1;
16 pthread_create(&threads[i], NULL, worker, &thread_ids[i]);
17 }
18 for (int i = 0; i < 3; i++) {
19 pthread_join(threads[i], NULL);
20 }
21 printf("Main thread: All threads have finished.\n");
22 return 0;
23 }
```

Program 2: Demonstrating a Race Condition


```

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     return 0;
19 }

```

The screenshot shows a Windows IDE (VS Code) with a project named "1151-LABS [WSL: UBUNTU]". The Explorer pane on the left lists files: q1.c, q2.c, q3.c, q4.1.c, q4.2.c, q5.c, and q5.out. The main editor shows the code for q5.c, which is the same as the code block above. The terminal at the bottom shows the command "gcc q5.c -o q5.out -lpthread" and the output of the program, which matches the expected and actual counter values.

```

esha@ESHA-DELL:~/1151-lab5$ code q5.c
esha@ESHA-DELL:~/1151-lab5$ gcc q5.c -o q5.out -lpthread
esha@ESHA-DELL:~/1151-lab5$ ./q5.out
Expected counter value: 200000
Actual counter value: 123427
esha@ESHA-DELL:~/1151-lab5$

```