



**National Textile University**  
**Department of Computer Science**

Subject:

Operating System

---

Submitted to:

Dr. Nasir Mehmood

---

Submitted by:

Dawar Abbas

---

Reg number:

1144

---

# Lab 5

```
File Edit Selection View ... Lab5 [WSL: Ubuntu]
```

EXPLORER

- LAB5 [WSL: UBUNTU]
- Q1
- Q1.c

C Q1.c

```
14 main()
15 {
16     printf("Main thread starting...\n");
17     printf("Main Thread ID: %lu\n", pthread_self());
18     // Create a new thread
19     pthread_create(&thread_id, NULL, thread_function, NULL);
20     // Wait for the thread to finish
21     pthread_join(thread_id, NULL);
22     printf("Main thread exiting...\n");
23     return 0;
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
bash - Lab5
```

```
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Q1.c -o Q1 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./Q1
Main thread starting...
Main Thread ID: 130946750203712
Hello from the new thread!
Thread ID: 130946747594432
Main thread exiting...
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

WSL: Ubuntu 0 0 0 0 Ln 17, Col 59 (4 selected) Spaces: 4 UTF-8 LF {} C Linux 2:31 AM 10/17/2025

```
File Edit Selection View ... Lab5 [WSL: Ubuntu]
```

EXPLORER

- LAB5 [WSL: UBUNTU]
- Q1
- Q1.c
- Q2
- Q2.c

C Q2.c

```
1 #include <stdio.h>
2 #include <pthread.h>
3 void *print_number(void *arg)
4 {
5     // We know that we've passed an integer pointer
6     int num = *(int *)arg; // Cast void* back to int*
7     printf("Thread received number: %d\n", num);
8     printf("Square: %d\n", num * num);
9     return NULL;
10 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
bash - Lab5
```

```
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./Q1
Main Thread ID: 130946750203712
Hello from the new thread!
Thread ID: 130946747594432
Main thread exiting...
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Q2.c -o Q2 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./Q2
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

WSL: Ubuntu 0 0 0 0 Ln 7, Col 30 Spaces: 4 UTF-8 LF {} C Linux 2:33 AM 10/17/2025

The image shows a Windows Subsystem for Linux (WSL) environment. The Explorer sidebar on the left displays the file structure of the 'LABS [WSL: UBUNTU]' directory, including files 'Q1.c', 'Q2', 'Q2A.c', 'Q3.c', and 'Q4.c'. The main editor window shows the source code for 'Q2A.c', which includes a function 'print\_number' and a 'main' function. The terminal output shows the successful compilation of 'Q2A.c' into 'Q2' and its execution, displaying the results of the program's calculations and thread creation.

```
Q2A.c > print_number(void *)
1  #include <stdio.h>
2
3  void *print_number(void *arg)
4  {
5      // We know that we've passed an integer pointer
6      float num = *(float *)arg; // Cast void* back to int*
7      printf("Thread received CGPA: %f\n", num);
8      printf("Double CGPA: %f\n", num * num);
9      return NULL;
10 }
11 int main()
12 {
13     pthread_t thread_id;
```

```
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./Q2
Square: 11.222499
Main thread done.
• dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Q2A.c -o Q2 -lpthread
• dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./Q2
Creating thread with argument: 3.350000
Thread received CGPA: 3.350000
Double CGPA: 11.222499
Main thread done.
• dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

The screenshot displays a Windows 11 desktop environment. A Windows taskbar is visible at the bottom, featuring the Start button, a search icon, and several pinned applications including File Explorer, Microsoft Edge, and the Visual Studio Code icon. The primary focus is the Visual Studio Code application, which is running a WSL (Windows Subsystem for Linux) instance of Ubuntu. The interface is divided into several panels: the Explorer panel on the left shows the file structure of the 'LAB5 [WSL: UBUNTU]' workspace, with files 'Q1.c', 'Q2.c', 'Q2A.c', 'Q3.c', and 'Q4.c'. The 'Q3.c' file is selected and its contents are displayed in the main editor. The code is a C program that demonstrates thread synchronization using a mutex. It defines a 'ThreadData' struct with an 'id' and a 'message' pointer. A 'printData' function is defined to print the thread's ID and message. The 'main' function creates two 'ThreadData' objects, 'data1' and 'data2', and spawns two threads, 'thread1' and 'thread2', using the 'pthread\_create' function. Each thread calls 'printData' to print its respective message. The 'main' function then calls 'pthread\_join' to wait for both threads to complete before printing 'All threads done.' and returning 0. The TERMINAL panel at the bottom shows the execution of the program. It starts with the command 'dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5\$ ./Q2', which outputs 'Thread received CGPA: 3.350000' and 'Double CGPA: 11.222499'. Then, the command 'dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5\$ gcc Q3.c -o Q3 -lpthread' is executed. Finally, the command 'dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5\$ ./Q3' is run, producing the output: 'Thread 1 says: Hello', 'Thread 2 says: World', and 'All threads done.'

```
File Edit Selection View ... Q Lab5 [WSL: Ubuntu]
```

EXPLORER

- LAB5 [WSL: UBUNTU]
  - Q1.c
  - Q2.c
  - Q2A.c
  - Q3
  - Q3.c
  - Q4.c

```
Q3.c > ThreadData
3 typedef struct
4 {
5     int id;
6     char *message;
7 } ThreadData;
8 void *printData(void *arg)
9 {
10     ThreadData *data = (ThreadData *)arg;
11     printf("Thread %d says: %s\n", data->id, data->message);
12     return NULL;
13 }
14 int main()
15 {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
bash - Lab5 + - - - - -
```

```
dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5$ ./Q2
Thread received CGPA: 3.350000
Double CGPA: 11.222499
Main thread done.
• dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5$ gcc Q3.c -o Q3 -lpthread
• dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5$ ./Q3
Thread 1 says: Hello
Thread 2 says: World
All threads done.
• dawar@DESKTOP-EG75J4:~/operating-system-1144/Lab5$
```

> OUTLINE  
> TIMELINE

WSL: Ubuntu 0 0 0 0

Ln 7, Col 14 Spaces: 4 UTF-8 LF {} C Linux

Upcoming Earnings

2:51 AM 10/17/2025

```
1 #include <stdio.h> //In this task we are passing multiple arguments using cgpa
2 #include <pthread.h>
3 typedef struct
4 {
5     float ThreadData::cgpa;
6     int id;
7     char *message;
8     float cgpa;
9 } ThreadData;
10 void *printData(void *arg)
11 {
12     ThreadData *data = (ThreadData *)arg;
13     printf("Thread %d name: %s cgpa: %f\n", data->id, data->message, data->cgpa);
14     return NULL;
15 }
16 int main()
17 {
18     pthread_t t1, t2;
19     ThreadData data1 = {1, "Dawar", 3.35};
20     pthread_create(&t1, NULL, printData, &data1);
21 }
```

bash - Lab5

```
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Task3(A).c -o task3 -lpthread
bash: syntax error near unexpected token '('
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Task3.c -o task3 -lpthread
^C
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Task3A.c -o task3 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./task3
Thread 1 name: Dawar cgpa: 3.350000
All threads done.
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

```
1 #include <stdio.h> // Returning result to the main thread
2 #include <pthread.h>
3 #include <stdlib.h>
4 void *calculate_sum(void *arg)
5 {
6     int n = *(int *)arg;
7     int *result = malloc(sizeof(int)); // Allocate memory for result
8     *result = 0;
9     for (int i = 1; i <= n; i++)
10     {
11         *result += i;
12     }
13     printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
14     return (void *)result; // Return the result
15 }
16 int main()
17 {
18 }
```

bash - Lab5

```
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./task3
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Task4.c -o task4 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./task4
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

```
File Edit Selection View ... Lab5 [WSL: Ubuntu]
EXPLORER
  LAB5 [WSL: UBUNTU]
    multiThread1
    C MultiThread1.c
    C Task1.c
    C Task2.c
    C Task2A.c
    C Task3.c
    C Task3A.c
    C Task4.c
  OUTLINE
  TIMELINE

C MultiThread1.c > main()
12 int main()
13 {
14     pthread_t threads[3];
15     int thread_ids[3];
16     for (int i = 0; i < 3; i++)
17     {
18         thread_ids[i] = i + 1;
19     }
20 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Lab5
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./task3
All threads done.
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc Task4.c -o task4 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./task4
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc MultiThread1.c -o multiThread1 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./multiThread1
Thread 1: Starting task...
Thread 3: Starting task...
Thread 2: Starting task...
Thread 3: Task completed!
Thread 2: Task completed!
Thread 1: Task completed!
Main thread: All threads have finished.
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```

```
File Edit Selection View ... Lab5 [WSL: Ubuntu]
EXPLORER
  LAB5 [WSL: UBUNTU]
    C MultiThread1.c
    C MultiThread2.c
    C Task1.c
    C Task2.c
    C Task2A.c
    C Task3.c
    C Task3A.c
    C Task4.c
  OUTLINE
  TIMELINE

C MultiThread2.c > increment(void *)
1 #include <stdio.h>
2 #include <pthread.h>
3 int counter = 0; // Shared variable
4 void *increment(void *arg)
5 {
6     for (int i = 0; i < 100000; i++)
7     {
8         counter++; // Not thread-safe
9     }
10 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Lab5
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc MultiThread1.c -o multiThread1 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./multiThread1
Thread 1: Starting task...
Thread 3: Starting task...
Thread 2: Starting task...
Thread 3: Task completed!
Thread 2: Task completed!
Thread 1: Task completed!
Main thread: All threads have finished.
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ gcc MultiThread2.c -o multiThread2 -lpthread
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$ ./multiThread2
Expected counter value: 200000
Actual counter value: 144008
dawar@DESKTOP-EG875J4:~/operating-system-1144/Lab5$
```