

National Textile University, Faisalabad

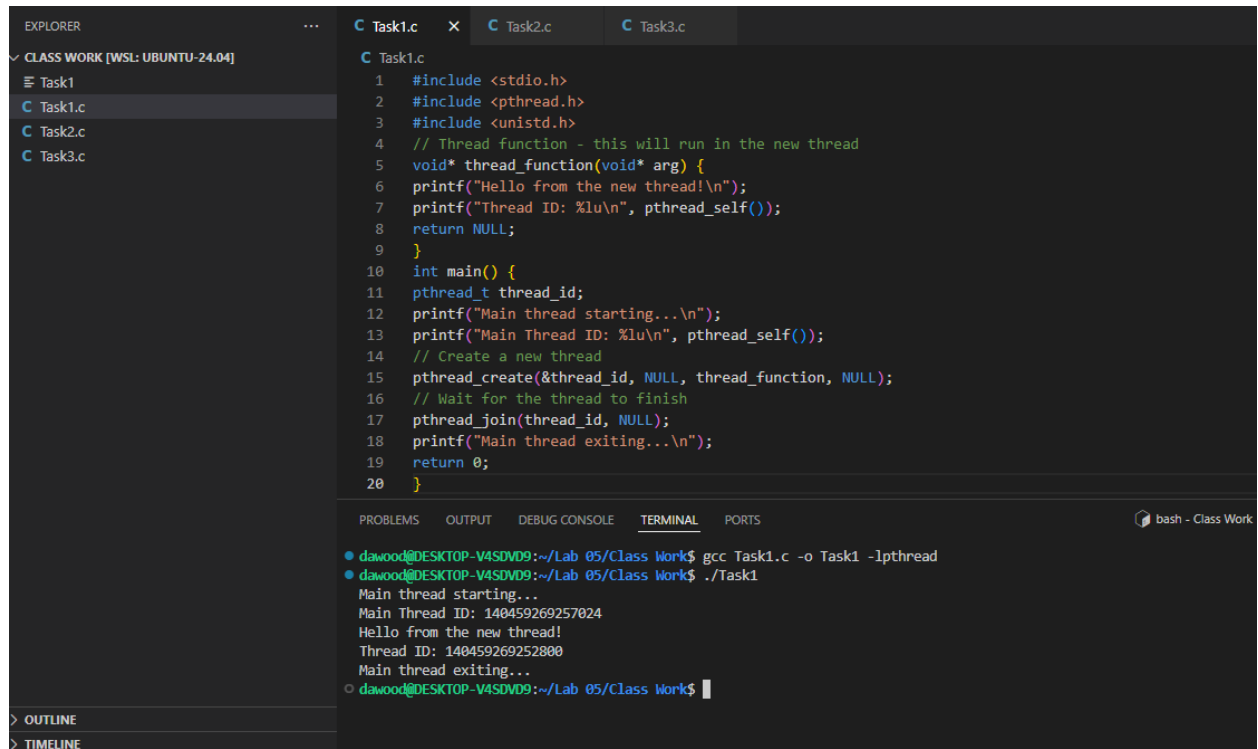


Department of Computer Science

| | |
|------------------------|------------------------|
| Name | Dawood Saif |
| Class | SE-5 th (A) |
| Reg. No. | 23-NTU-CS-1145 |
| Course | Operating system |
| Submitted To | Sir Nasir Mehmood |
| Submission Date | 17/10/2025 |
| Lab No. | 5 (Class Work) |

Lab No. 5 Operating system:

Program 1



The screenshot shows a code editor with three tabs: Task1.c, Task2.c, and Task3.c. The Explorer panel on the left shows the file structure under 'CLASS WORK [WSL: UBUNTU-24.04]'. The main editor window displays the code for Task1.c, which includes headers for stdio, pthread, and unistd. It defines a thread function and a main function that creates a new thread, prints messages, and waits for the thread to finish. The terminal at the bottom shows the compilation and execution of the program, with output messages from both the main thread and the newly created thread.

```
EXPLORER
CLASS WORK [WSL: UBUNTU-24.04]
  Task1
  Task1.c
  Task2.c
  Task3.c

Task1.c
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 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Class Work

● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ gcc Task1.c -o Task1 -lpthread
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ ./Task1
Main thread starting...
Main Thread ID: 140459269257024
Hello from the new thread!
Thread ID: 140459269252800
Main thread exiting...
○ dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$
```

Program2

```
CLASS WORK [WSL: UBUNTU-24.04]
Task1
Task1.c
Task2
Task2.c
Task3.c

Task2.c
1  #include <stdio.h>
2  #include <pthread.h>
3
4  void* print_number(void* arg)
5  {
6      // We know that we've passed a float pointer
7      float num = *(float*)arg; // Cast void* back to float*
8      printf("Thread received number: %f\n", num);
9      printf("Square: %f\n", num * num);
10     return NULL;
11 }
12
13 int main() {
14     pthread_t thread_id;
15     float number = 2.7;
16
17     printf("Creating thread with argument: %f\n", number);
18     // Pass address of 'number' to thread
19     pthread_create(&thread_id, NULL, print_number, &number);
20     pthread_join(thread_id, NULL);
21 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ gcc Task2.c -o Task2 -lpthread
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ ./Task2
Creating thread with argument: 2.700000
Thread received number: 2.700000
Square: 7.290000
Main thread done.
○ dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$
```

Program3

CLASS WORK [WSL: UBUNTU-24.04]

Task1
Task1.c
Task2
Task2.c
Task3
Task3.c

Task3.c

```
1  #include <stdio.h>
2  #include <pthread.h>
3  typedef struct {
4  int id;
5  char* message;
6  } ThreadData;
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 int main() {
13     pthread_t t1, t2;
14     ThreadData data1 = {1, "Dawood Saif"};
15     ThreadData data2 = {2, "My cgpa is 2.7"};
16
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 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ gcc Task3.c -o Task3 -lpthread
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ ./Task3
Thread 1 says: Dawood Saif
Thread 2 says: My cgpa is 2.7
All threads done.
○ dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$
```

Program4:

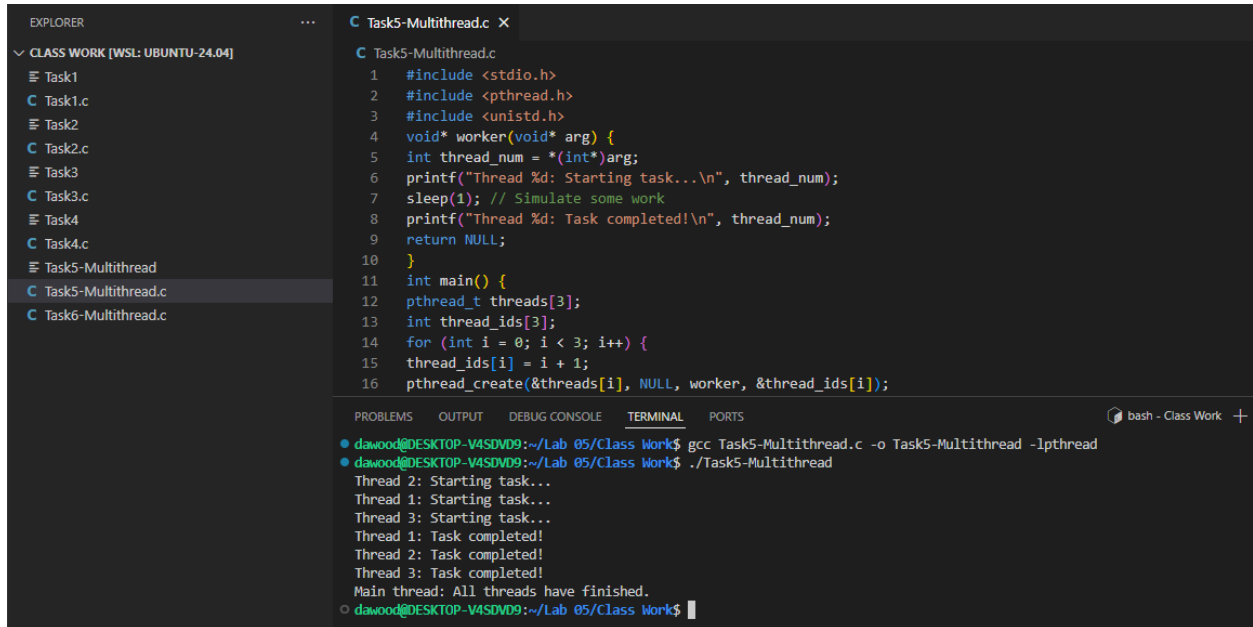
```
✓ CLASS WORK [WSL: UBUNTU-24.04]
  Task1
  C Task1.c
  Task2
  C Task2.c
  Task3
  C Task3.c
  Task4
  C Task4.c

C Task4.c
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);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ gcc Task4.c -o Task4 -lpthread
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ ./Task4
  Thread calculated sum of 1 to 100 = 5050
  Main received result: 5050
○ dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$
```

Now Program Examples of Multithreading

Program 1: Creating and Running Multiple Threads



The screenshot displays a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'CLASS WORK [WSL: UBUNTU-24.04]' containing files Task1, Task1.c, Task2, Task2.c, Task3, Task3.c, Task4, Task4.c, Task5-Multithread, Task5-Multithread.c, and Task6-Multithread.c. The code editor shows the content of Task5-Multithread.c, which is a C program that creates and runs three threads. The terminal at the bottom shows the compilation and execution of the program, with output indicating the start and completion of each thread.

```
EXPLORER
...
CLASS WORK [WSL: UBUNTU-24.04]
  Task1
  Task1.c
  Task2
  Task2.c
  Task3
  Task3.c
  Task4
  Task4.c
  Task5-Multithread
  Task5-Multithread.c
  Task6-Multithread.c

C Task5-Multithread.c X
C Task5-Multithread.c
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 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Class Work +
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ gcc Task5-Multithread.c -o Task5-Multithread -lpthread
● dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$ ./Task5-Multithread
Thread 2: Starting task...
Thread 1: Starting task...
Thread 3: Starting task...
Thread 1: Task completed!
Thread 2: Task completed!
Thread 3: Task completed!
Main thread: All threads have finished.
○ dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work$
```

Program 2: Demonstrating a Race Condition

```
C Task5-Multithread.c  C Task6-Multithread.c X
C Task6-Multithread.c
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 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work\$ gcc Task6-Multithread.c -o Task6-Multithread -lpthread
- dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work\$./Task6-Multithread
Expected counter value: 200000
Actual counter value: 129351
- dawood@DESKTOP-V4SDVD9:~/Lab 05/Class Work\$