



National Textile University

**Department of Computer Science**

Subject:  
**Operating System**

---

Submitted to:  
**Nasir Mahmood**

---

Submitted by:  
**Afia Maham**

---

Reg number:  
**23-NTU-CS-1126**

---

Lab no:  
**05**

---

Semester:  
**5<sup>th</sup>**

**Task 01:**

```
#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

// Thread function - this will run in the new thread
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());

    // Create a new thread
    pthread_create(&thread_id, NULL, thread_function, NULL);

    // Wait for the thread to finish
    pthread_join(thread_id, NULL);
    printf("Main thread exiting...\n");
    return 0;
}
```

The screenshot shows a Visual Studio Code editor window with the title bar 'Operating System [WSL: Ubuntu-24.04]'. The Explorer sidebar on the left shows a project structure with folders 'HackerRank', 'HackerRank-1', 'HackerRank-2', 'Lab-4', and 'Lab-5'. Under 'Lab-5', there are files 'lab5-task01.c' and 'lab5-task02.c'. The main editor area displays the code for 'lab5-task01.c':

```
1 #include <pthread.h>
2 #include <unistd.h>
3 // Thread function - this will run in the new thread
4 void* thread_function(void* arg)
5 {
6     printf("Hello from the new thread!\n");
7     printf("Thread ID: %lu\n", pthread_self());
8     return NULL;
9 }
10
11 int main()
12 {
13     pthread_t thread_id;
14     // ...
```

The TERMINAL panel at the bottom shows the execution of the program:

```
afiamaham@DESKTOP-190QPQH:~/Operating System$ gcc Lab-5/lab5-task01.c -o Lab-5/lab5-task01-out
afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task01-out
Main thread starting...
Main Thread ID: 140380925994816
Hello from the new thread!
Thread ID: 140380925990592
Main thread exiting...
afiamaham@DESKTOP-190QPQH:~/Operating System$
```

## Task 02:

```
#include <stdio.h>
```

```
#include <pthread.h>
```

```
void* print_number(void* arg){
```

```
    // We know that we've passed an integer pointer
```

```
    int num = *(int*)arg; // Cast void* back to int*
```

```
    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);
```

```

// Pass address of 'number' to thread

pthread_create(&thread_id, NULL, print_number, &number);

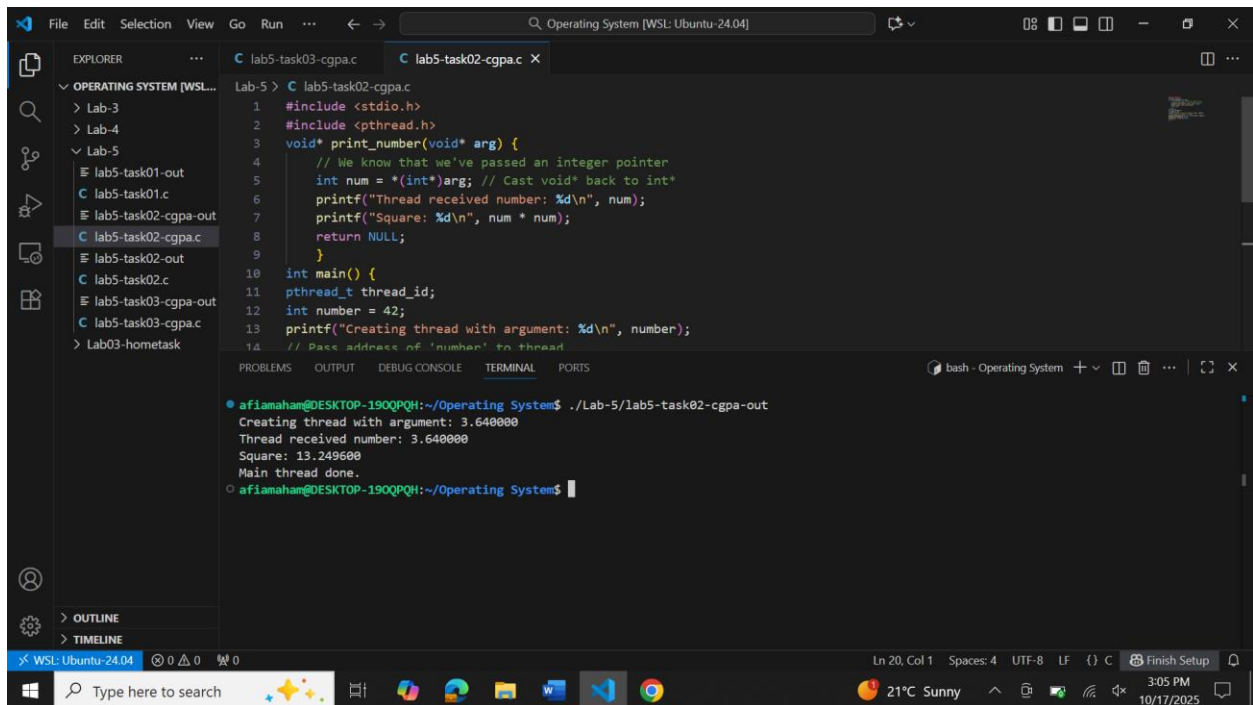
pthread_join(thread_id, NULL);

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

return 0;

}

```



The screenshot shows a Visual Studio Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure for 'Lab-5' with files like 'lab5-task01.c', 'lab5-task02-cgpa.c', and 'lab5-task02.c'. The main editor window displays the code for 'lab5-task02-cgpa.c', which includes `<stdio.h>` and `<pthread.h>`. It defines a `print_number` function that takes an integer pointer, prints the number and its square, and returns NULL. The `main` function creates a thread with `pthread_create`, passes the address of a variable `number` (initialized to 42), and joins the thread with `pthread_join`. The terminal output shows the program execution: 'Creating thread with argument: 3.640000', 'Thread received number: 3.640000', 'Square: 13.249600', and 'Main thread done.'.

```

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

```

```

afiamahan@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task02-cgpa-out
Creating thread with argument: 3.640000
Thread received number: 3.640000
Square: 13.249600
Main thread done.
afiamahan@DESKTOP-190QPQH:~/Operating System$

```

### Task 03:

```

#include <stdio.h>

#include <pthread.h>

typedef struct {

    float cgpa;

    char* name;

} ThreadData;

```

```

void* printData(void* arg) {

    ThreadData* data = (ThreadData*)arg;

    printf("Thread %f says: %s\n", data->cgpa, data->name);

    return NULL;

}

int main() {

    pthread_t t1;

    ThreadData data1 = {3.64, "Afia"};

    pthread_create(&t1, NULL, printData, &data1);

    pthread_join(t1, NULL);

    printf("All threads done.\n");

    return 0;

}

```

The screenshot shows a Visual Studio Code editor window with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The file `lab5-task03-cgpa.c` is selected.
- EDITOR:** The main workspace showing the source code of `lab5-task03-cgpa.c`. The code is a C program that uses `pthread` to create a thread that prints the CGPA and name of a student.
- TERMINAL:** A terminal window at the bottom showing the execution of the program. The output is:
 

```

afiamaham@DESKTOP-190QPQH:~/Operating System$ gcc Lab-5/lab5-task03-cgpa.c -o Lab-5/lab5-task03-cgpa-out
afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task03-cgpa-out
Thread 3.640000 says: Afia
All threads done.
afiamaham@DESKTOP-190QPQH:~/Operating System$

```

**Task04:**

```
#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

void* calculate_sum(void* arg) {

    int n = *(int*)arg;

    int* result = malloc(sizeof(int)); // Allocate memory for result

    *result = 0;

    for (int i = 1; i <= n; i++) {

        *result += i;

    }

    printf("Thread calculated sum of 1 to %d = %d\n", n, *result);

    return (void*)result; // Return the result

}

int main() {

    pthread_t thread_id;

    int n = 100;

    void* sum;

    pthread_create(&thread_id, NULL, calculate_sum, &n);

    // Get the return value from thread

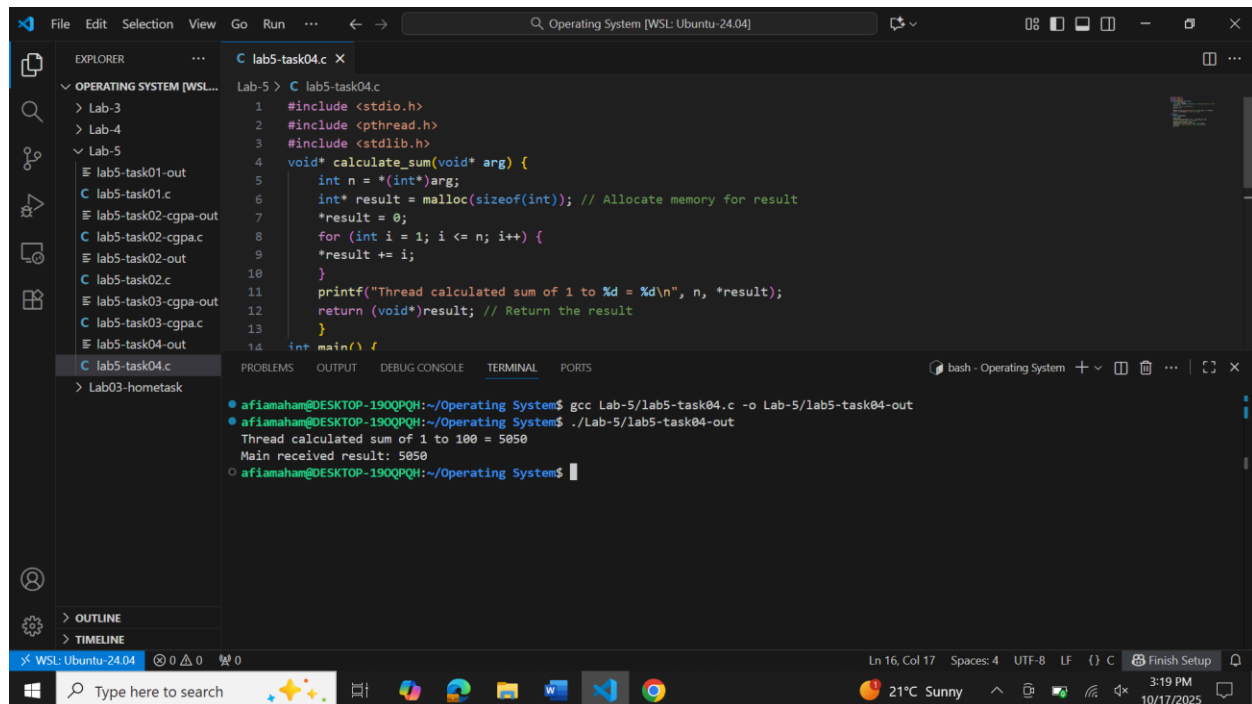
    pthread_join(thread_id, &sum);

    printf("Main received result: %d\n", *(int*)sum);

    free(sum); // Don't forget to free allocated memory

    return 0;

}
```



```
File Edit Selection View Go Run ... Operating System [WSL: Ubuntu-24.04]
EXPLORER
OPERATING SYSTEM [WSL: Ubuntu-24.04]
  Lab-3
  Lab-4
  Lab-5
    lab5-task01-out
    lab5-task01.c
    lab5-task02-cgpa-out
    lab5-task02-cgpa.c
    lab5-task02-out
    lab5-task02.c
    lab5-task03-cgpa-out
    lab5-task03-cgpa.c
    lab5-task04-out
    lab5-task04.c
    Lab03-hometask
  OUTLINE
  TIMELINE
C lab5-task04.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() {
    afiamaham@DESKTOP-190QPQH:~/Operating System$ gcc Lab-5/lab5-task04.c -o Lab-5/lab5-task04-out
    afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task04-out
    Thread calculated sum of 1 to 100 = 5050
    Main received result: 5050
    afiamaham@DESKTOP-190QPQH:~/Operating System$
```

## Task 05:

```
#include <stdio.h>
```

```
#include <pthread.h>
```

```
#include <unistd.h>
```

```
void* worker(void* arg) {
```

```
    int thread_num = *(int*)arg;
```

```
    printf("Thread %d: Starting task...\n", thread_num);
```

```
    sleep(1); // Simulate some work
```

```
    printf("Thread %d: Task completed!\n", thread_num);
```

```
    return NULL;
```

```
}
```

```
int main() {
```

```

pthread_t threads[3];

int thread_ids[3];

for (int i = 0; i < 3; i++) {

    thread_ids[i] = i + 1;

    pthread_create(&threads[i], NULL, worker, &thread_ids[i]);

}

for (int i = 0; i < 3; i++) {

    pthread_join(threads[i], NULL);

}

printf("Main thread: All threads have finished.\n");

return 0;

}

```

The screenshot displays the Visual Studio Code interface with a C program named `lab5-task05.c` open. The program is designed to create and manage three threads. The `worker` function takes a thread ID as an argument, prints a message, sleeps for 1 second, prints another message, and then returns `NULL`. The `main` function initializes an array of `pthread_t` types, an array of thread IDs, and then creates three threads using `pthread_create`. After creating the threads, it uses `pthread_join` to wait for each thread to complete. Finally, it prints a message indicating that all threads have finished and returns 0.

The terminal output shows the execution of the program. It starts with the compilation command: `gcc Lab-5/lab5-task05.c -o Lab-5/lab5-task05-out`. The execution output shows the following sequence of messages:

```

Thread 1: Starting task...
Thread 3: Starting task...
Thread 2: Starting task...
Thread 1: Task completed!
Thread 3: Task completed!
Thread 2: Task completed!
Main thread: All threads have finished.

```

The status bar at the bottom indicates the file is in the `Lab-5` directory, and the system tray shows the date and time as 3:33 PM on 10/17/2025.



**Task06:**

```
#include <stdio.h>

#include <pthread.h>

int counter = 0; // Shared variable

void* increment(void* arg) {
    for (int i = 0; i < 100000; i++) {
        counter++; // Not thread-safe
    }
    return NULL;
}

int main() {
    pthread_t t1, t2;
    pthread_create(&t1, NULL, increment, NULL);
    pthread_create(&t2, NULL, increment, NULL);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    printf("Expected counter value: 200000\n");
    printf("Actual counter value: %d\n", counter);
    return 0;
}
```

File Edit Selection View Go Run ... ← → Operating System [WSL: Ubuntu-24.04]

EXPLORER ... C lab5-task05.c C lab5-task06.c X

OPERATING SYSTEM [WSL: Ubuntu-24.04]

- > Lab-3
- > Lab-4
- > Lab-5
  - lab5-task01-out
  - C lab5-task01.c
  - lab5-task02-cgpa-out
  - C lab5-task02-cgpa.c
  - lab5-task02-out
  - C lab5-task02.c
  - lab5-task03-cgpa-out
  - C lab5-task03-cgpa.c
  - lab5-task04-out
  - C lab5-task04.c
  - lab5-task05-out
  - C lab5-task05.c
  - lab5-task06-out
  - C lab5-task06.c
- > Lab03-hometask

OUTLINE

TIMELINE

Lab-5 > C lab5-task06.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);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - Operating System + - | | | |

```
afiamaham@DESKTOP-190QPQH:~/Operating System$ gcc Lab-5/lab5-task06.c -o Lab-5/lab5-task06-out
afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 126993
afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 148865
afiamaham@DESKTOP-190QPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 144431
afiamaham@DESKTOP-190QPQH:~/Operating System$
```

Ln 20, Col 1 Spaces: 4 UTF-8 LF {} C Finish Setup

21°C Sunny 3:46 PM 10/17/2025