National Textile University

Department of Computer Science

Subject: Operating System

Submitted to: Sir Nasir

Submitted by: Maha

Reg. number: 23-NTU-CS-1170

Lab 4 (class task)

Semester:5th

3. C Programs with Threads

Program 1: Creating a Simple Thread

Objective: Create a thread and print messages from both main thread and new thread.

Code:

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
void* fun(void* arg) {
   printf("Lab 4 class task of threads \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, fun, NULL);

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

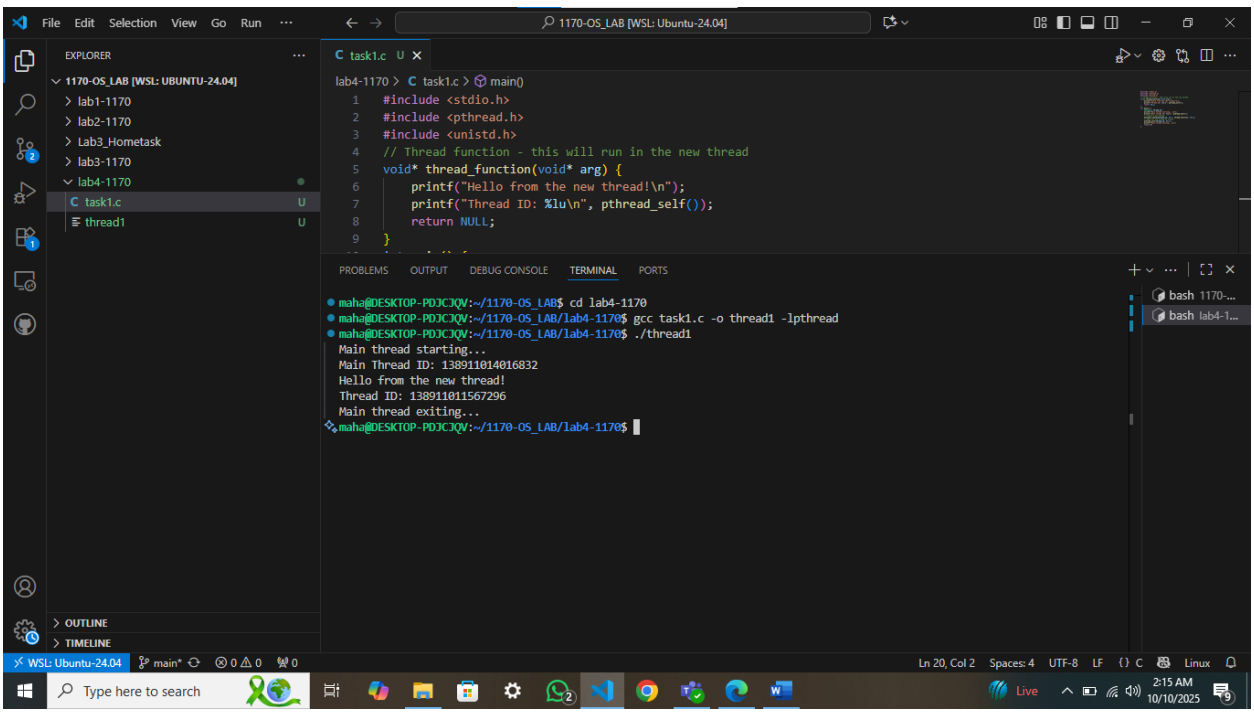
```
    return 0;
}
```

Compilation:

Compile and run:

```
gcc thread1.c -o thread1 -lpthread
./thread1
```

Terminal:



Program 2: Passing Arguments to Threads

Objective: Pass data to a thread function.

Code:

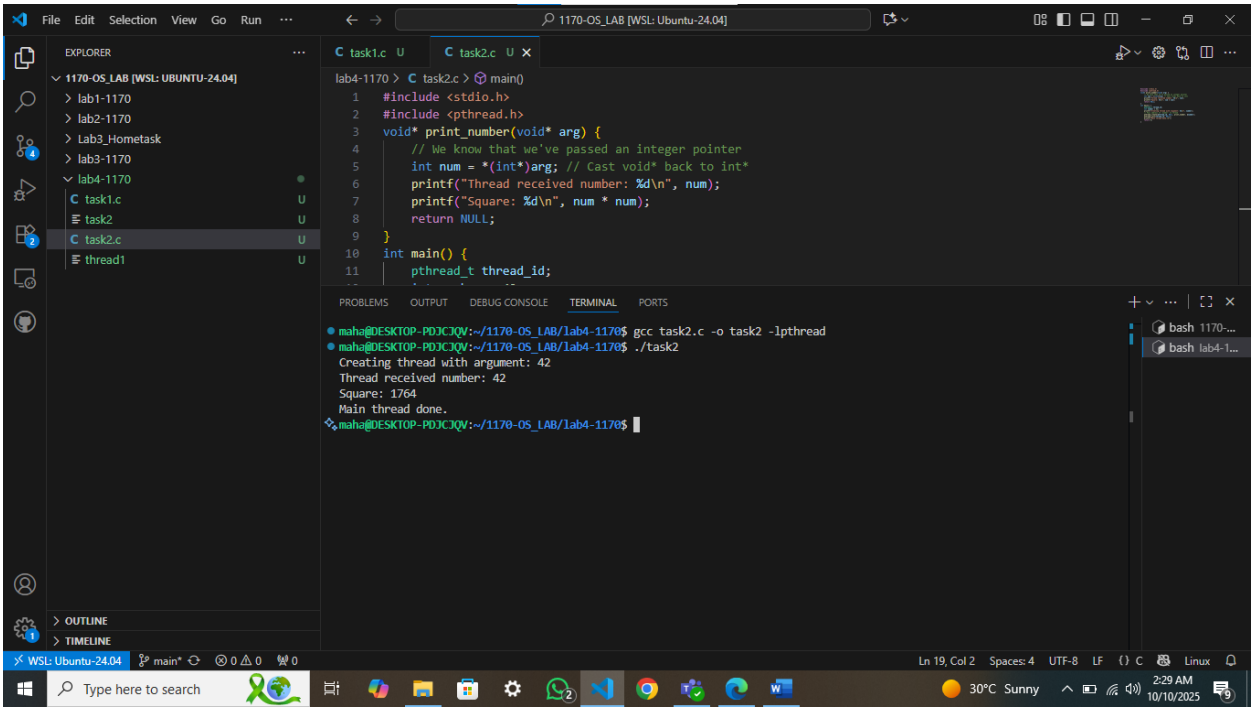
```
#include <stdio.h>

#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;
}
```

Compile and run:

```
gcc thread2.c -o thread2 -lpthread
./thread2
```

Terminal:



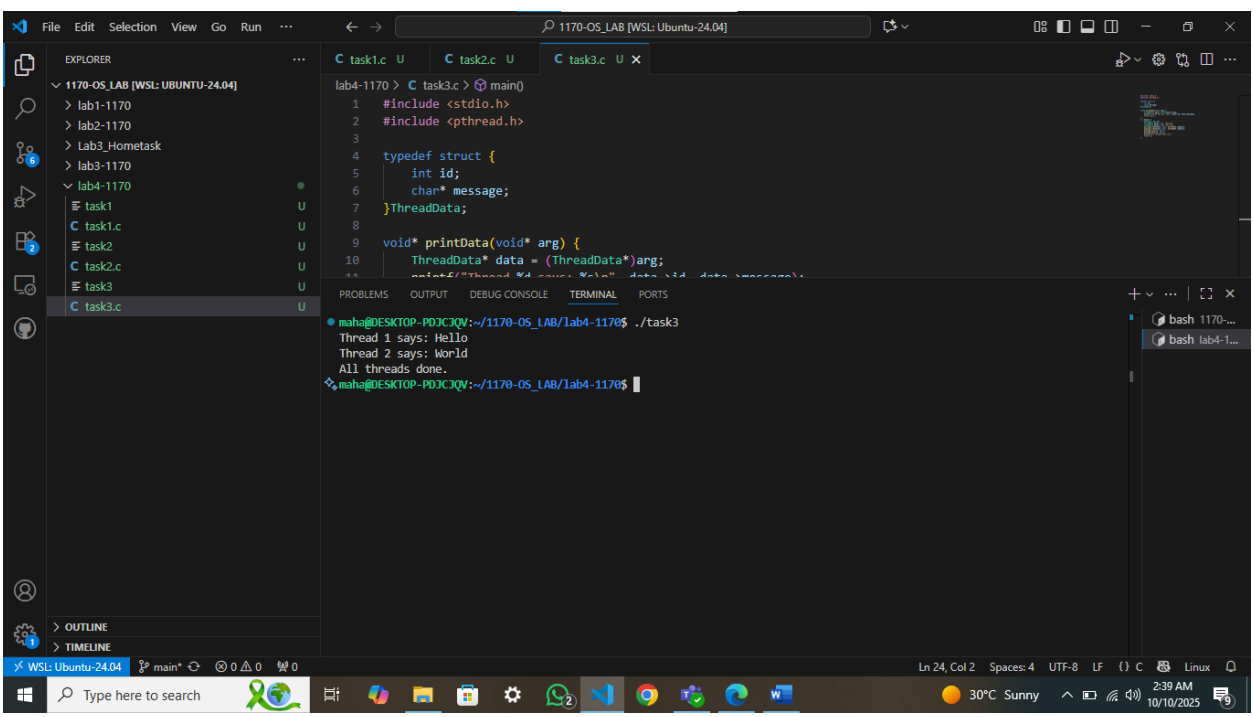
Program 3: Passing Multiple Data

Code:

```
#include <stdio.h>

#include <pthread.h>


typedef struct {

    int id;

    char* message;

}ThreadData;


void* printData(void* arg) {

    ThreadData* data = (ThreadData*)arg;

    printf("Thread %d says: %s\n", data->id, data->message);
```

```
    return NULL;
}
int main() {
    pthread_t t1, t2;
    ThreadData data1 = {1, "Hello"};
    ThreadData data2 = {2, "World"};
    pthread_create(&t1, NULL, printData, &data1);
    pthread_create(&t2, NULL, printData, &data2);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    printf("All threads done.\n");
    return 0;
}
```

Terminal:



Program 4: Multiple Threads

Objective: Create multiple threads executing the same function.

Code:

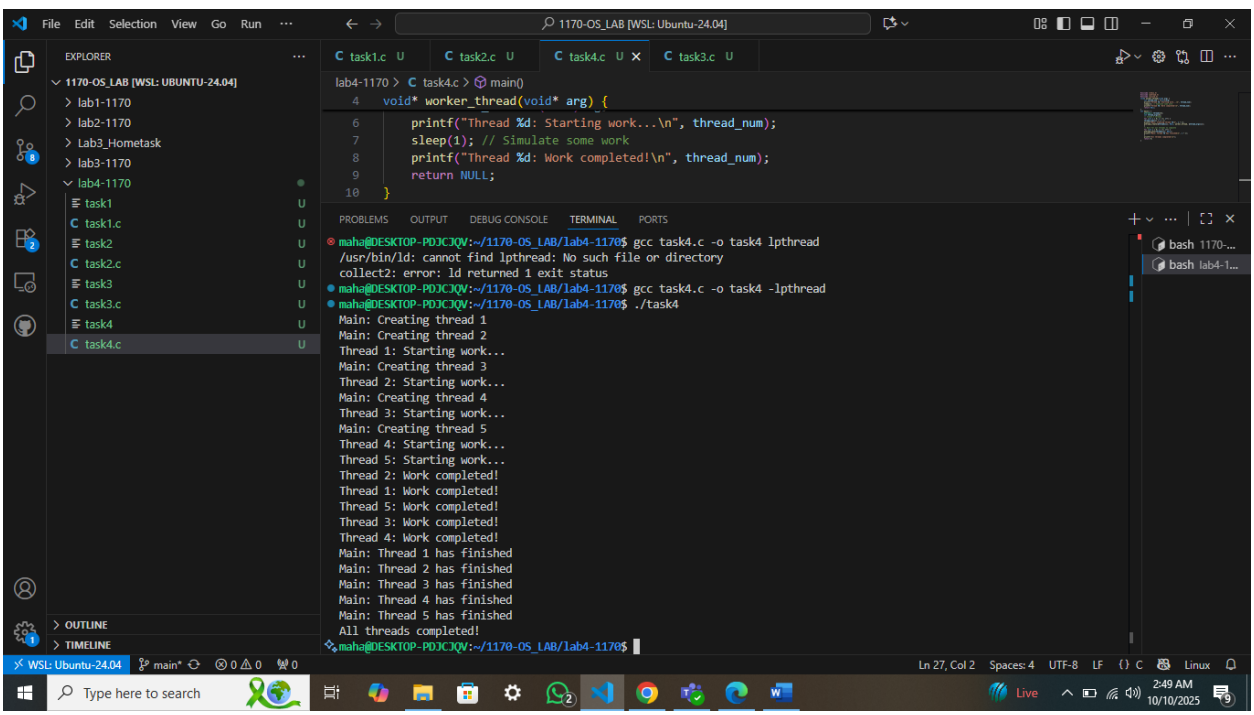
```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* worker_thread(void* arg) {
   int thread_num = *(int*)arg;
   printf("Thread %d: Starting work...\n", thread_num);
   sleep(1); // Simulate some work
   printf("Thread %d: Work completed!\n", thread_num);
   return NULL;
}
int main() {
   pthread_t threads[5];
   int thread_args[5];
   // Create 5 threads
   for (int i = 0; i < 5; i++) {
   thread_args[i] = i + 1;
   printf("Main: Creating thread %d\n", i + 1);
   pthread_create(&threads[i], NULL, worker_thread, &thread_args[i]);
   }
   // Wait for all threads to complete
   for (int i = 0; i < 5; i++) {
```

```
    pthread_join(threads[i], NULL);
    printf("Main: Thread %d has finished\n", i + 1);
    }
    printf("All threads completed!\n");
    return 0;
}
```

Compile and run:

```
gcc thread3.c -o thread3 -lpthread
./thread3
```

Terminal:



Question 5:

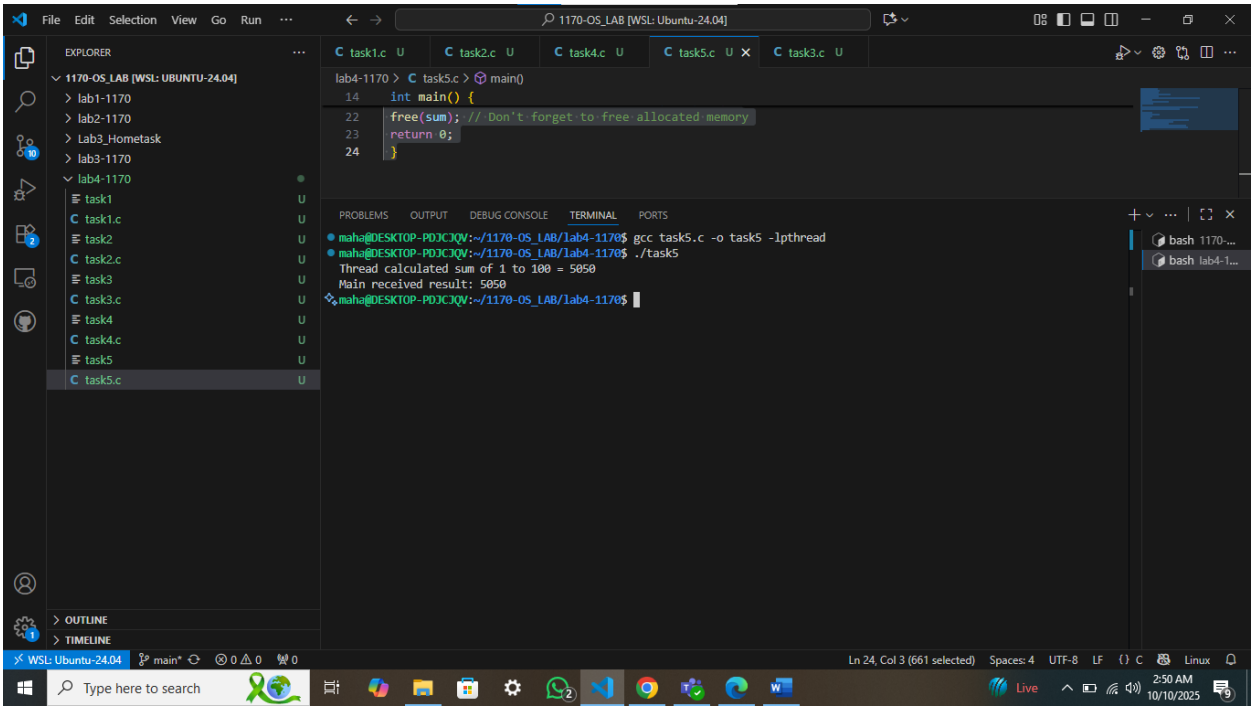
Program 5: Thread Return Values

Code:

```
#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;
```

```
}
```

Terminal:


File Edit Selection View Go Run
1170-OS_LAB [WSL: Ubuntu-24.04]
EXPLORER
1170-OS_LAB [WSL: UBUNTU-24.04]
lab1-1170
lab2-1170
Lab3_Hometask
lab3-1170
lab4-1170
task1
task1.c
task2
task2.c
task3
task3.c
task4
task4.c
task5
task5.c
OUTLINE
TIMELINE
task1.c U
task2.c U
task4.c U
task5.c U
task3.c U
lab4-1170 > task5.c > main()
int main() {
free(sum); // Don't forget to free allocated memory
return 0;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/lab4-1170$ gcc task5.c -o task5 -lpthread
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/lab4-1170$ ./task5
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/lab4-1170$
bash 1170-...
bash lab4-1...
WSL: Ubuntu-24.04
main*
Ln 24, Col 3 (661 selected)
Spaces: 4
UTF-8
LF
C
Linux
Type here to search
Live
2:50 AM
10/10/2025