# National Textile University

# Department of Computer Science

**Subject:**

Operating System

**Submitted to:**

Sir Naseer

**Submitted by:**

Ayema

**Reg number:**

23-NTU-CS-1142

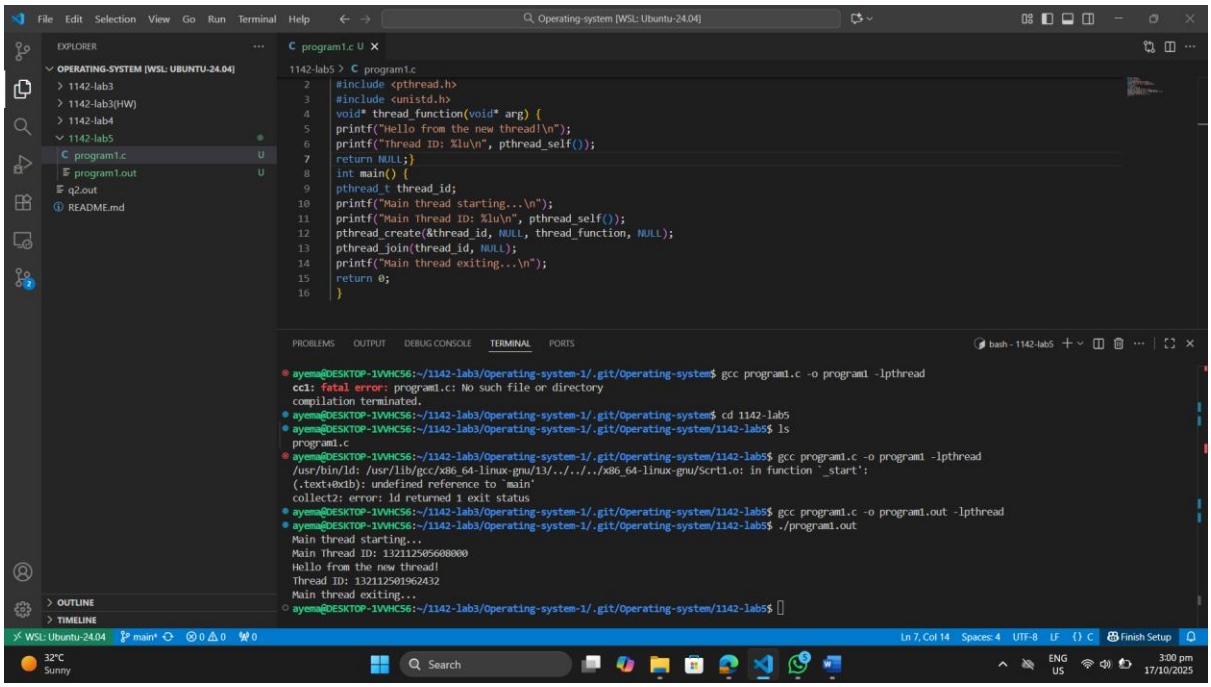
**Lab no. :**

05

**Semester:** 5Th

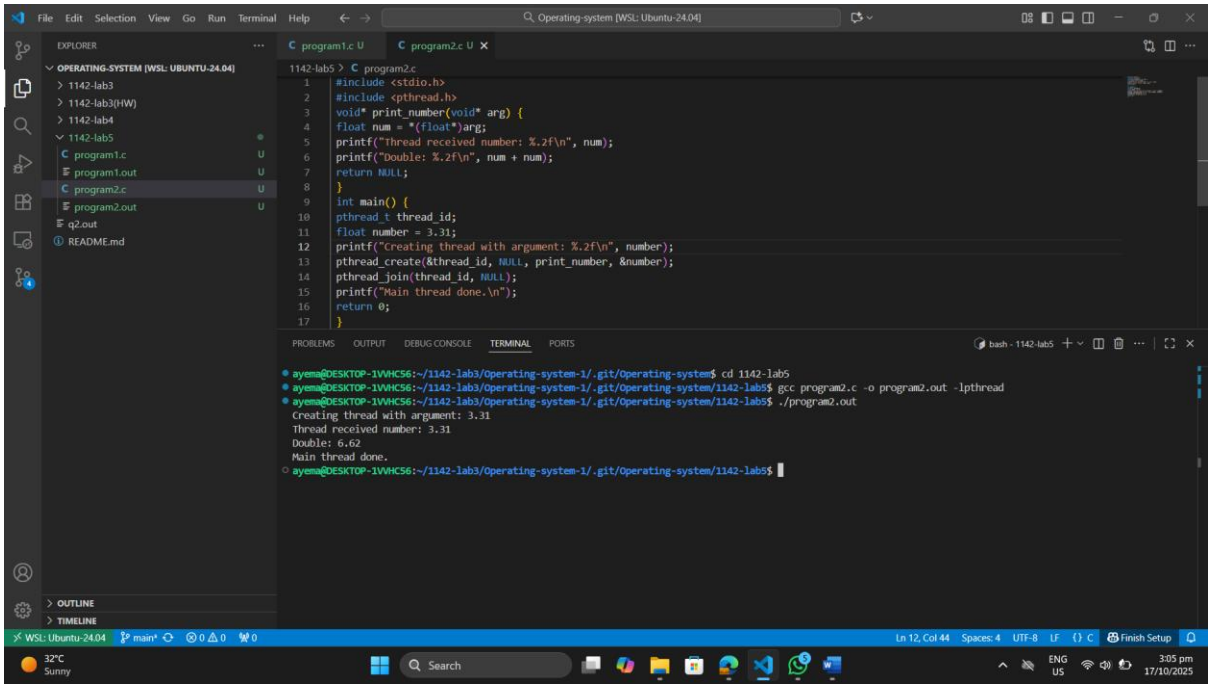
# Program 1: Creating a Simple Thread

```c
#include <stdio.h>
 #include <pthread.h>
 #include <unistd.h>
 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());
 pthread_create(&thread_id, NULL, thread_function, NULL);
 pthread_join(thread_id, NULL);
 printf("Main thread exiting...\n");
 return 0;
 }
```



## Program 2: Passing Arguments to Threads

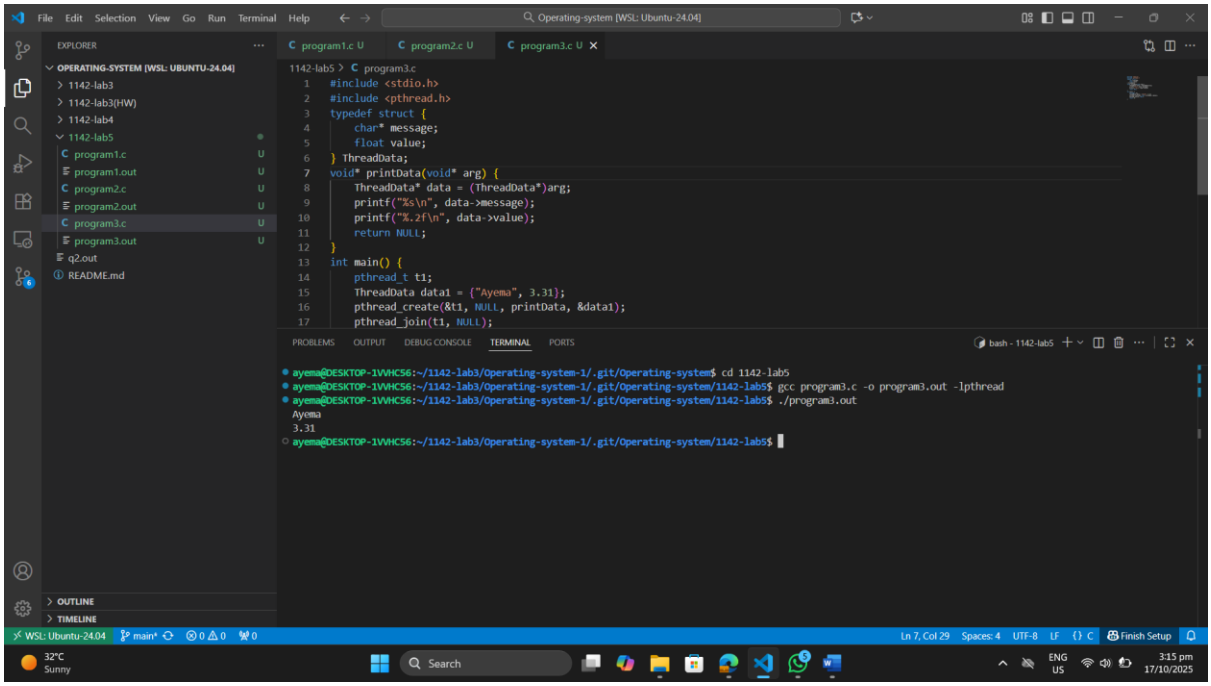
```c
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
float num = *(float*)arg;
printf("Thread received number: %.2f\n", num);
printf("Double: %.2f\n", num + num);
return NULL;
}
int main() {
pthread_t thread_id;
float number = 3.31;
printf("Creating thread with argument: %.2f\n", number);
pthread_create(&thread_id, NULL, print_number, &number);
pthread_join(thread_id, NULL);
printf("Main thread done.\n");
return 0;
}
```



## Program 3: Passing Multiple Data:

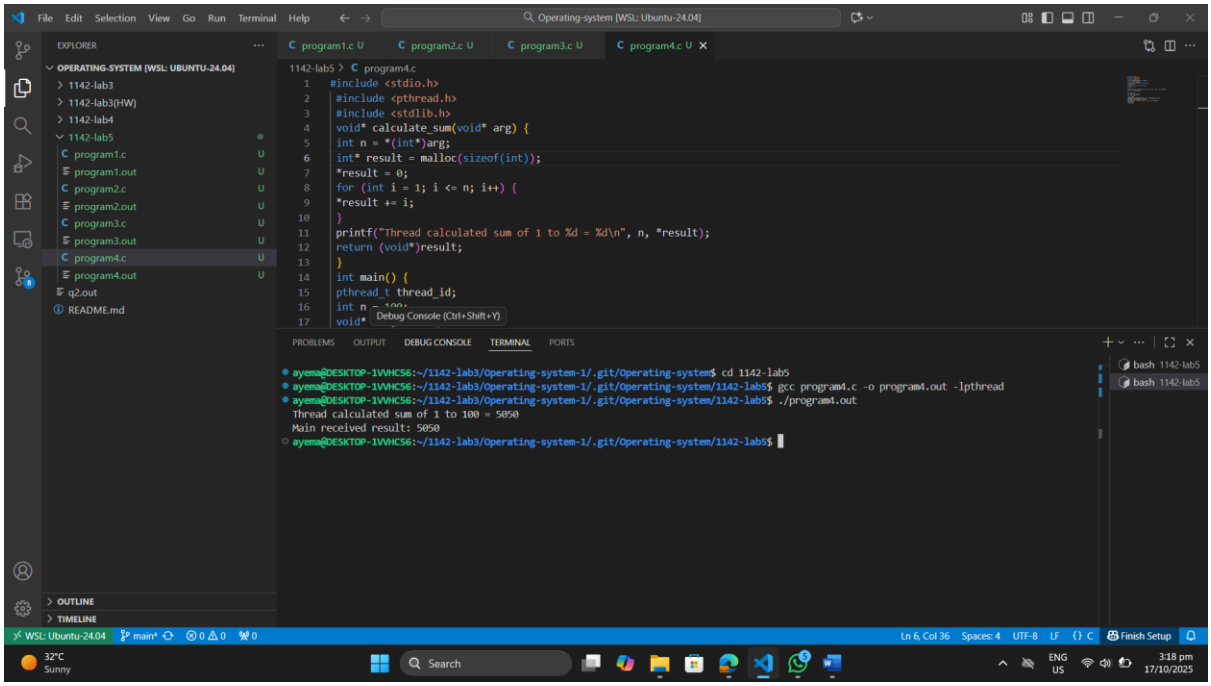
```c
#include <stdio.h>
#include <pthread.h>
typedef struct {
    char* message;
    float value;
} ThreadData;
void* printData(void* arg) {
    ThreadData* data = (ThreadData*)arg;
    printf("%s\n", data->message);
    printf("%.2f\n", data->value);
    return NULL;
}
int main() {
    pthread_t t1;
    ThreadData data1 = {"Ayema", 3.31};
    pthread_create(&t1, NULL, printData, &data1);
    pthread_join(t1, NULL);
    return 0;
}

```



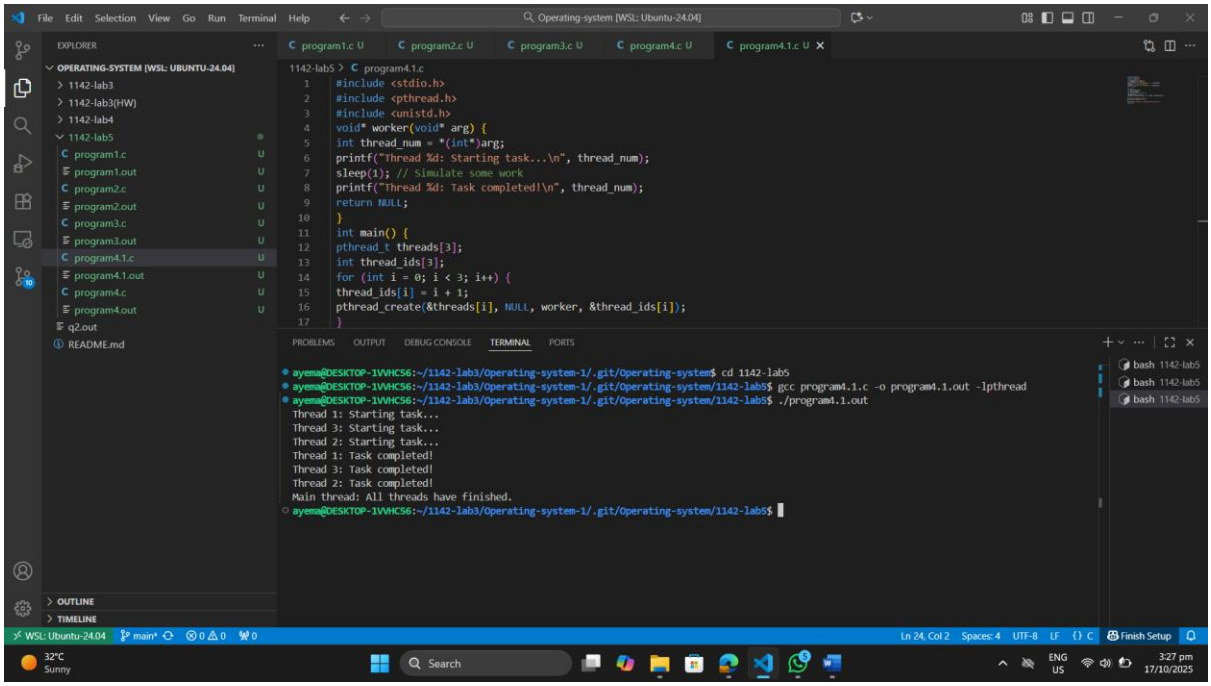

## Program 4: Thread Return Values

```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
int n = *(int*)arg;
int* result = malloc(sizeof(int));
*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;
}
int main() {
pthread_t thread_id;
int n = 100;
void* sum;
pthread_create(&thread_id, NULL, calculate_sum, &n);
pthread_join(thread_id, &sum);
printf("Main received result: %d\n", *(int*)sum);
free(sum);
return 0;
}
```



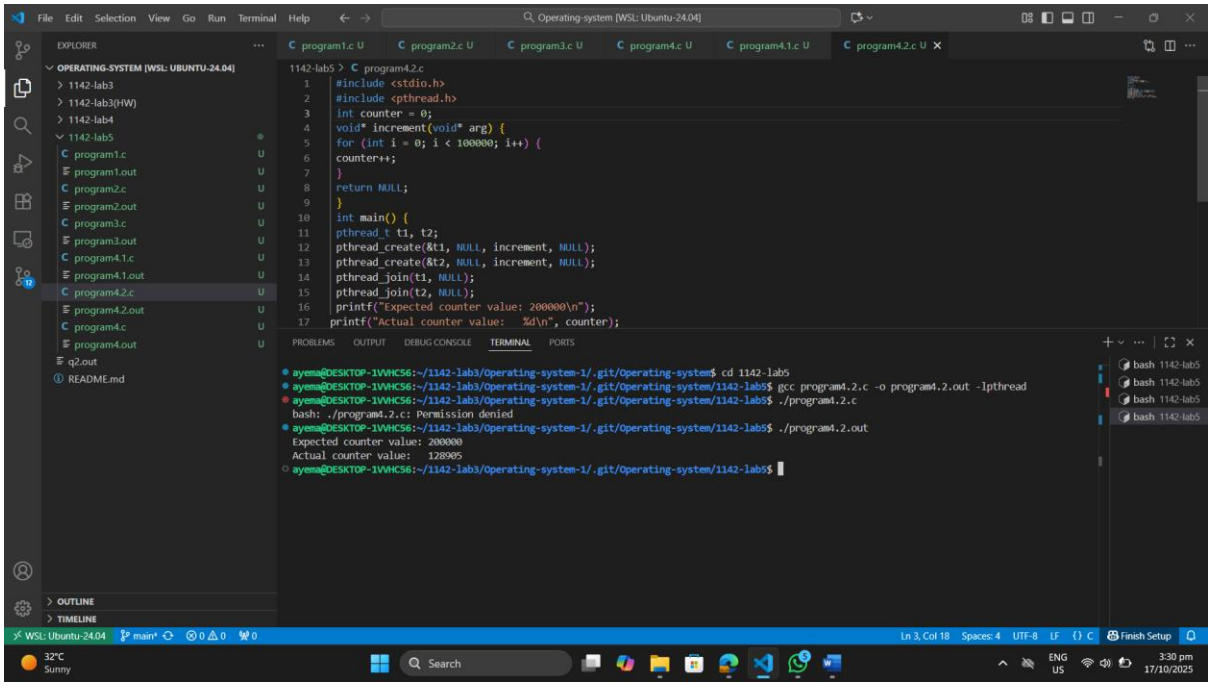

## Program 4.1: Creating and Running Multiple Threads

```c
#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;
}
```




## Program 4.2: Demonstrating a Race Condition

```c
#include <stdio.h>
#include <pthread.h>
int counter = 0;
void* increment(void* arg) {
for (int i = 0; i < 100000; i++) {
counter++;
}
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 Terminal Help
Operating-system [WSL: Ubuntu-24.04]
EXPLORER
OPERATING-SYSTEM [WSL: UBUNTU-24.04]
1142-lab3
1142-lab3(HW)
1142-lab4
1142-lab5
program1.c
program1.out
program2.c
program2.out
program3.c
program3.out
program4.1.c
program4.1.out
program4.2.c
program4.2.out
program4.c
program4.out
q2.out
README.md
OUTLINE
TIMELINE
program1.c program2.c program3.c program4.c program4.1.c program4.2.c
1142-lab5 > program4.2.c
#include <stdio.h>
#include <pthread.h>
int counter = 0;
void* increment(void* arg) {
for (int i = 0; i < 100000; i++) {
counter++;
}
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);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ayema@DESKTOP-1VVHC56:~/1142-lab3/Operating-system-1/.git/Operating-system$ cd 1142-lab5
ayema@DESKTOP-1VVHC56:~/1142-lab3/Operating-system-1/.git/Operating-system/1142-lab5$ gcc program4.2.c -o program4.2.out -lpthread
ayema@DESKTOP-1VVHC56:~/1142-lab3/Operating-system-1/.git/Operating-system/1142-lab5$ ./program4.2.c
bash: ./program4.2.c: Permission denied
ayema@DESKTOP-1VVHC56:~/1142-lab3/Operating-system-1/.git/Operating-system/1142-lab5$ ./program4.2.out
Expected counter value: 200000
Actual counter value:  128905
ayema@DESKTOP-1VVHC56:~/1142-lab3/Operating-system-1/.git/Operating-system/1142-lab5$
bash 1142-lab5
bash 1142-lab5
bash 1142-lab5
bash 1142-lab5
WSL: Ubuntu-24.04 main* 0 0 0
Ln 3, Col 18 Spaces: 4 UTF-8 LF C Finish Setup
32°C Sunny
Search
ENG US
3:30 pm 17/10/2025