National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir Mehmood

Submitted by:

Areeba Tariq

Reg number:

23-NTU-CS-1139

Lab no:
4TH

Semester:

5th

CODE

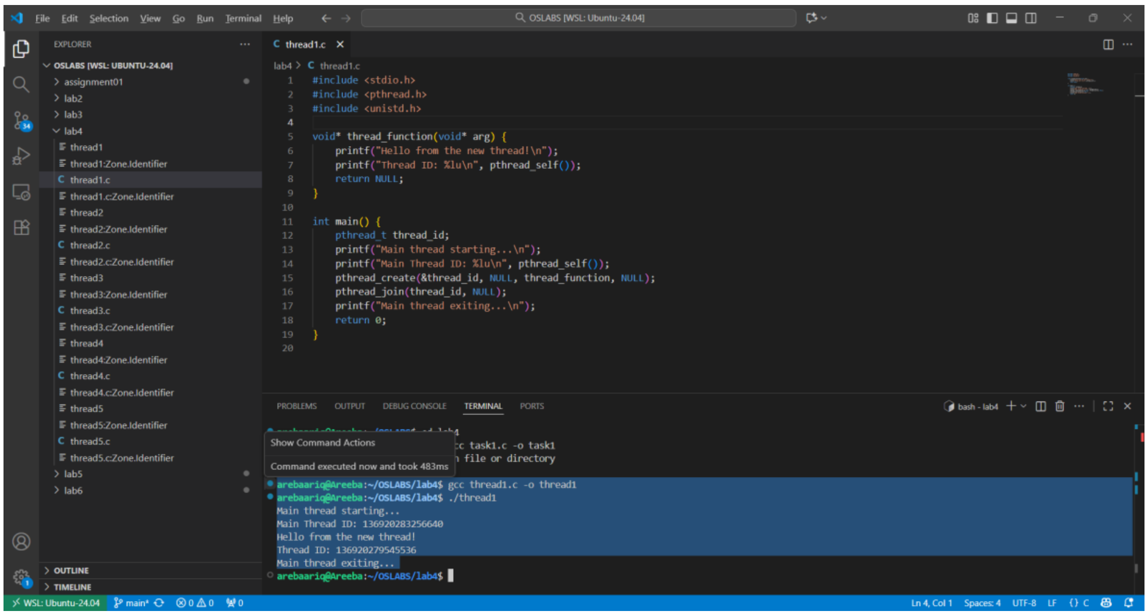
TASK1

```
#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;
}
```

OUTPUT


File Edit Selection View Go Run Terminal Help
OSLABS [WSL: Ubuntu-24.04]
EXPLORER
OSLABS [WSL: UBUNTU-24.04]
assignment01
lab2
lab3
lab4
thread1
thread1:Zone.Identifier
thread1.c
thread1.c:Zone.Identifier
thread2
thread2:Zone.Identifier
thread2.c
thread2.c:Zone.Identifier
thread3
thread3:Zone.Identifier
thread3.c
thread3.c:Zone.Identifier
thread4
thread4:Zone.Identifier
thread4.c
thread4.c:Zone.Identifier
thread5
thread5:Zone.Identifier
thread5.c
thread5.c:Zone.Identifier
lab5
lab6
OUTLINE
TIMELINE
thread1.c
lab4 > thread1.c
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - lab4
Show Command Actions
Command executed now and took 483ms
WSL: Ubuntu-24.04 main* 0 0 0
Ln 4, Col 1 Spaces: 4 UTF-8 LF {} 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;
}
```

```
cc task1.c -o task1
 file or directory
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread1.c -o thread1
arebaariq@Areeba:~/OSLABS/lab4$ ./thread1
Main thread starting...
Main Thread ID: 136920283256640
Hello from the new thread!
Thread ID: 136920279545536
Main thread exiting...
arebaariq@Areeba:~/OSLABS/lab4$
```

REMARKS

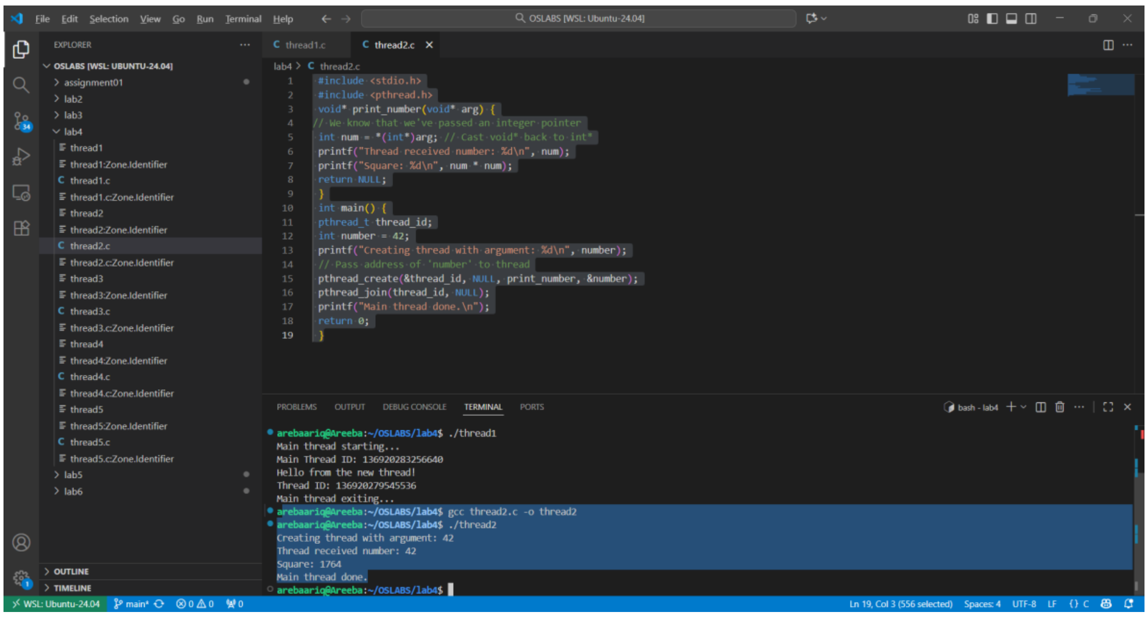
Thread successfully created and joined. Shows how main waits for thread completion.

TASK2

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;
}
```

OUTPUT



```
#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;
}
```

```
arebaariq@Areeba:~/OSLABS/lab4$ ./thread1
Main thread starting...
Main Thread ID: 136920283256640
Hello from the new thread!
Thread ID: 136920279545536
Main thread exiting...
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread2.c -o thread2
arebaariq@Areeba:~/OSLABS/lab4$ ./thread2
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
arebaariq@Areeba:~/OSLABS/lab4$
```

REMARKS

Demonstrates argument passing to a thread using pointer and thread execution behavior.

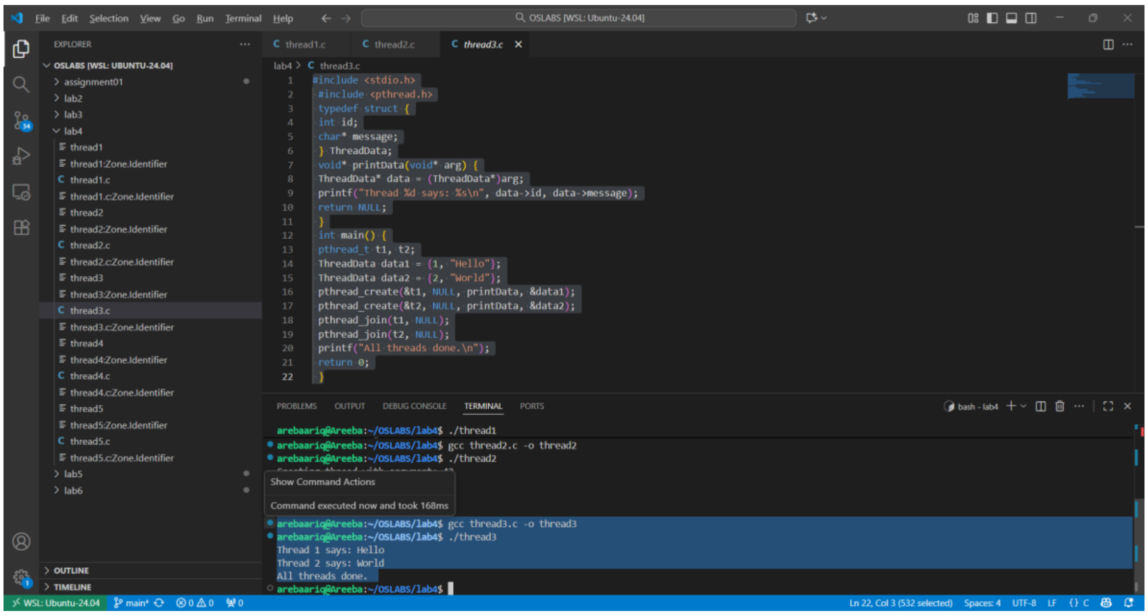
TASK3

CODE

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

```c
#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;
}
```

OUTPUT



```c
#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;
}
```

```
arebaariq@Areeba:~/OSLABS/lab4$ ./thread1
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread2.c -o thread2
arebaariq@Areeba:~/OSLABS/lab4$ ./thread2
Show Command Actions
Command executed now and took 168ms
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread3.c -o thread3
arebaariq@Areeba:~/OSLABS/lab4$ ./thread3
Thread 1 says: Hello
Thread 2 says: World
All threads done.
arebaariq@Areeba:~/OSLABS/lab4$
```

File Edit Selection View Go Run Terminal Help
OSLABS [WSL: Ubuntu-24.04]
EXPLORER
thread1.c
thread2.c
thread3.c
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - lab4
WSL: Ubuntu-24.04
Ln 22, Col 3 (532 selected) Spaces: 4 UTF-8 LF {} C

REMARKS

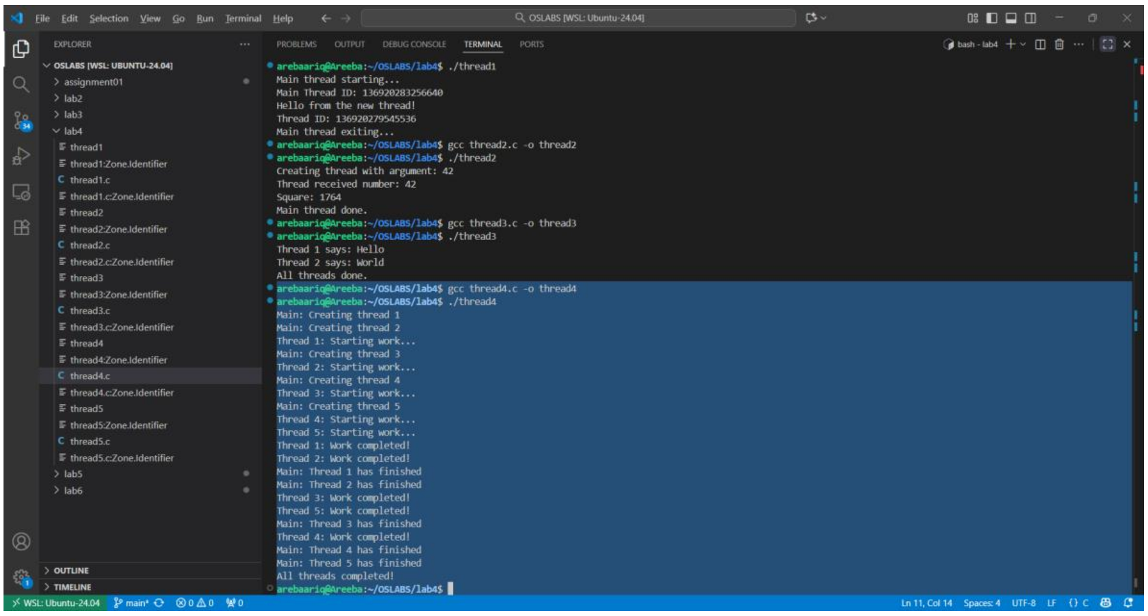
Shows struct usage in thread communication. Both threads execute independently.

TASK4

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;
 }
```

OUTPUT



```
arebaariq@Areeba:~/OSLABS/lab4$ ./thread1
Main thread starting...
Main Thread ID: 136920283256640
Hello from the new thread!
Thread ID: 136920279545536
Main thread exiting...
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread2.c -o thread2
arebaariq@Areeba:~/OSLABS/lab4$ ./thread2
Creating thread with argument: 42
Thread received number: 42
Square: 1764
Main thread done.
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread3.c -o thread3
arebaariq@Areeba:~/OSLABS/lab4$ ./thread3
Thread 1 says: Hello
Thread 2 says: World
All threads done.
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread4.c -o thread4
arebaariq@Areeba:~/OSLABS/lab4$ ./thread4
Main: Creating thread 1
Main: Creating thread 2
Thread 1: Starting work...
Main: Creating thread 3
Thread 2: Starting work...
Main: Creating thread 4
Thread 3: Starting work...
Main: Creating thread 5
Thread 4: Starting work...
Thread 5: Starting work...
Thread 1: Work completed!
Thread 2: Work completed!
Main: Thread 1 has finished
Main: Thread 2 has finished
Thread 3: Work completed!
Thread 5: Work completed!
Main: Thread 3 has finished
Thread 4: Work completed!
Main: Thread 4 has finished
Main: Thread 5 has finished
All threads completed!
arebaariq@Areeba:~/OSLABS/lab4$
```

REMARKS

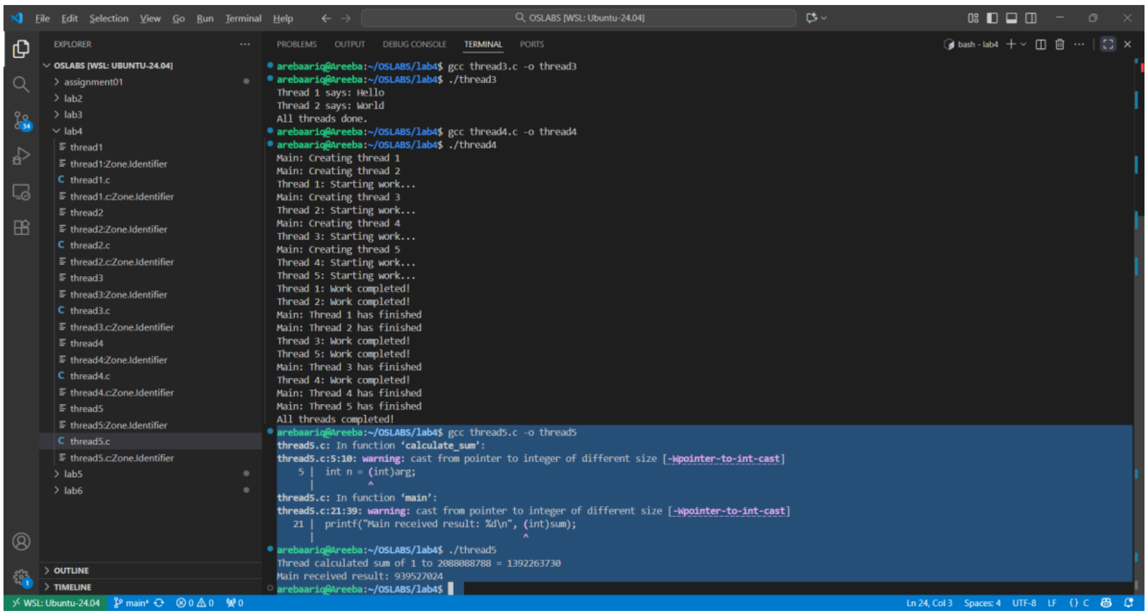
Demonstrates multiple thread creation, synchronization using pthread_join, and parallel behavior**.**

TASK5

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;
 }
```

OUTPUT



```
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread3.c -o thread3
arebaariq@Areeba:~/OSLABS/lab4$ ./thread3
Thread 1 says: Hello
Thread 2 says: World
All threads done.
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread4.c -o thread4
arebaariq@Areeba:~/OSLABS/lab4$ ./thread4
Main: Creating thread 1
Main: Creating thread 2
Thread 1: Starting work...
Main: Creating thread 3
Thread 2: Starting work...
Main: Creating thread 4
Thread 3: Starting work...
Main: Creating thread 5
Thread 4: Starting work...
Thread 5: Starting work...
Thread 1: Work completed!
Thread 2: Work completed!
Main: Thread 1 has finished
Main: Thread 2 has finished
Thread 3: Work completed!
Thread 5: Work completed!
Main: Thread 3 has finished
Thread 4: Work completed!
Main: Thread 4 has finished
Main: Thread 5 has finished
All threads completed!
arebaariq@Areeba:~/OSLABS/lab4$ gcc thread5.c -o thread5
thread5.c: In function ‘calculate_sum’:
thread5.c:5:10: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
    5 |  int n = (int)arg;
      |          ^
thread5.c: In function ‘main’:
thread5.c:21:39: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
   21 |  printf("Main received result: %d\n", (int)sum);
      |                                       ^
arebaariq@Areeba:~/OSLABS/lab4$ ./thread5
Thread calculated sum of 1 to 2088088788 = 1392263730
Main received result: 939527024
arebaariq@Areeba:~/OSLABS/lab4$
```

REMARKS

Shows thread return value using dynamic memory, retrieved with pthread_join.