National Textile University

Department of Computer Science

Subject:
Operating System

Submitted to:
Nasir Mahmood

Submitted by:
Afia Maham

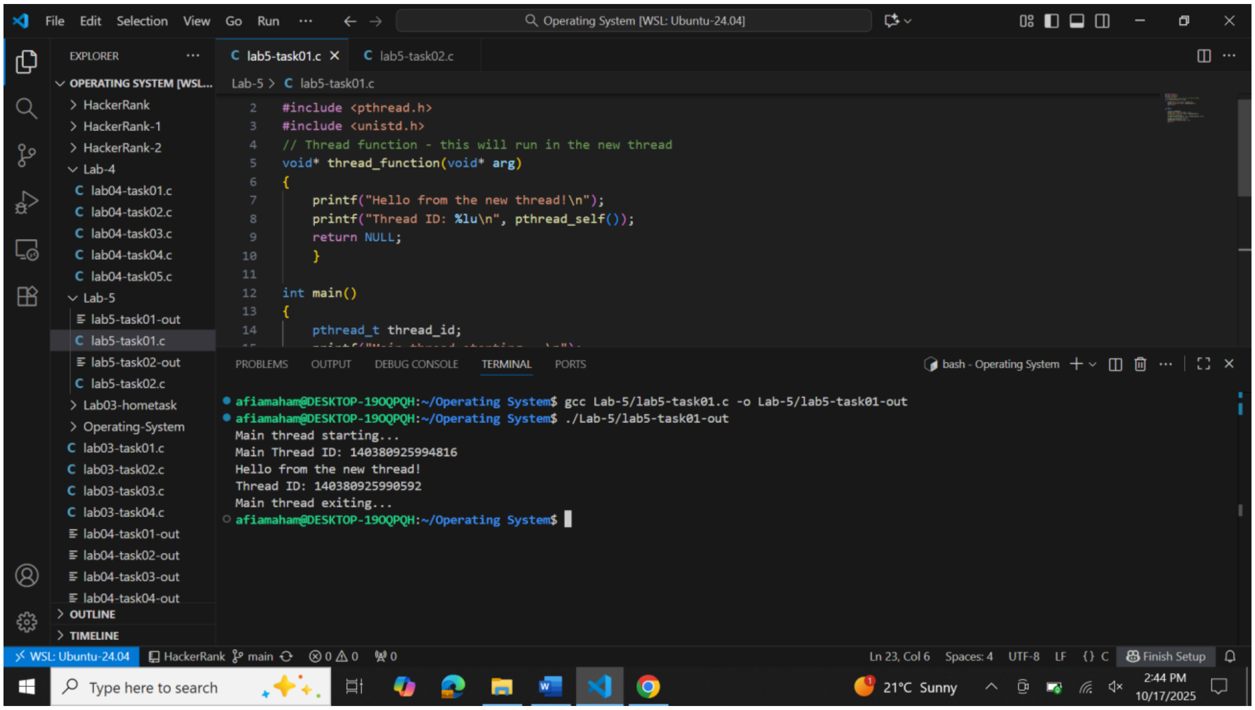
Reg number:
23-NTU-CS-1126

Lab no:
05

Semester:
5th

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;
  }
```

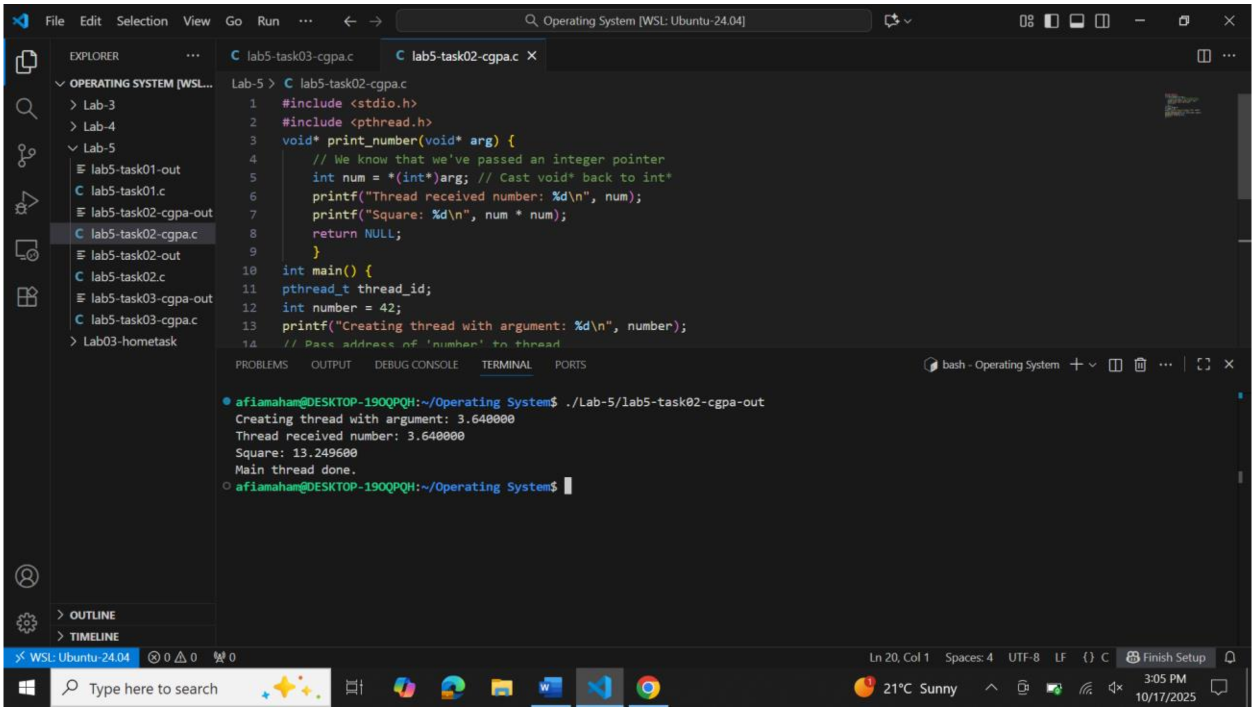


```
#include <pthread.h>
#include <unistd.h>
// Thread function - this will run in the new thread
void* thread_function(void* arg)
{
    printf("Hello from the new thread!\n");
    printf("Thread ID: %lu\n", pthread_self());
    return NULL;
    }

int main()
{
    pthread_t thread_id;
```

```
afiamaham@DESKTOP-19OQPQH:~/Operating System$ gcc Lab-5/lab5-task01.c -o Lab-5/lab5-task01-out
afiamaham@DESKTOP-19OQPQH:~/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-19OQPQH:~/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;

}
```



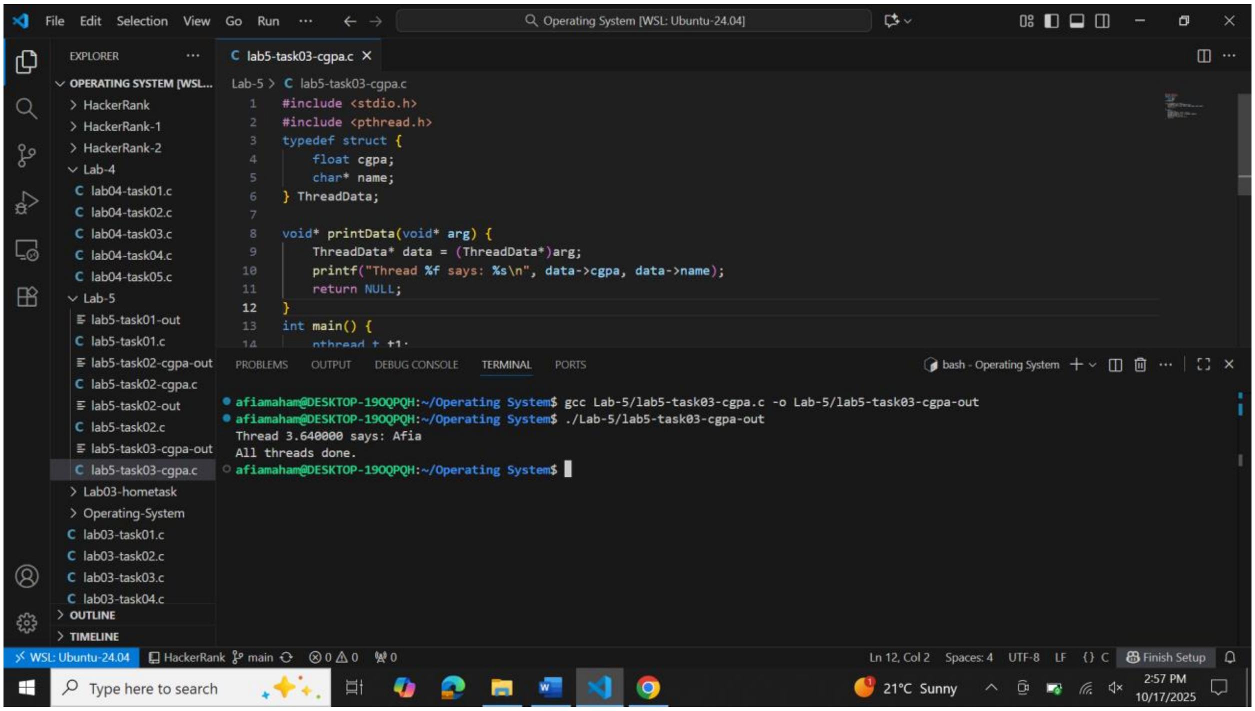
```c
#include <stdio.h>
#include <pthread.h>
void* print_number(void* arg) {
    // We know that we've passed an integer pointer
    int num = *(int*)arg; // Cast void* back to int*
    printf("Thread received number: %d\n", num);
    printf("Square: %d\n", num * num);
    return NULL;
    }
int main() {
pthread_t thread_id;
int number = 42;
printf("Creating thread with argument: %d\n", number);
// Pass address of 'number' to thread
```

```
afiamaham@DESKTOP-19OQPQH:~/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.
afiamaham@DESKTOP-19OQPQH:~/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;
}
```

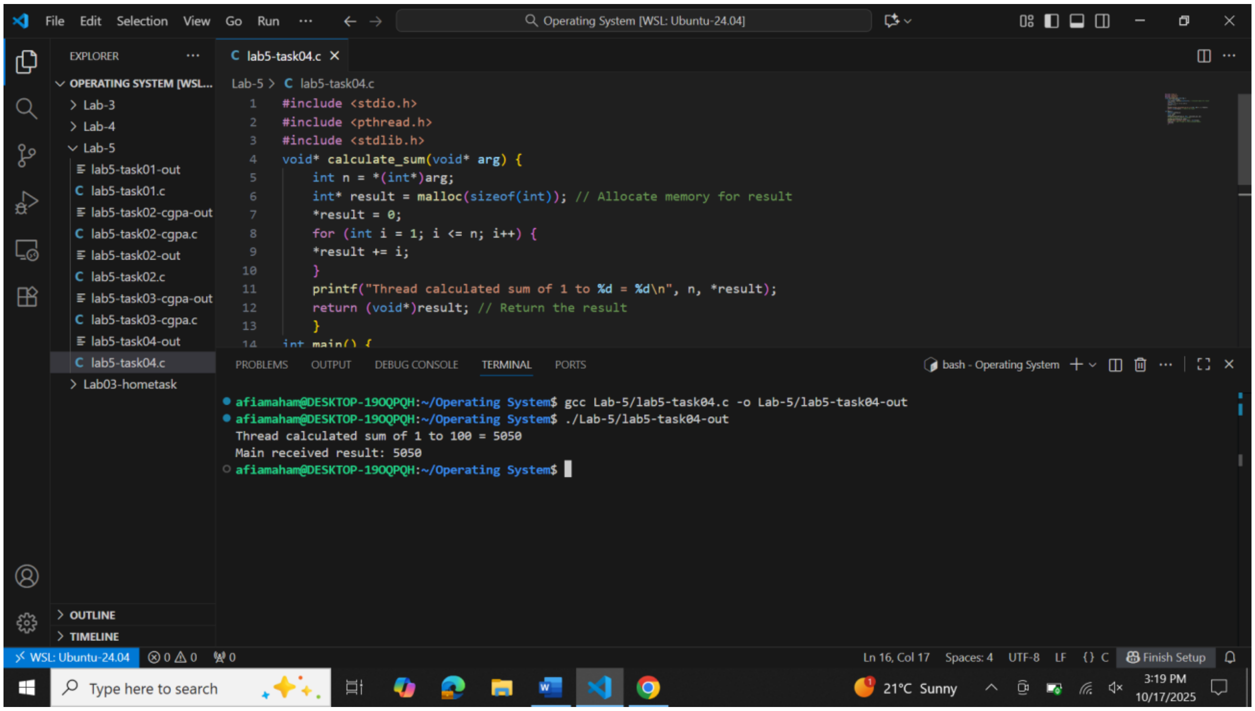


```
#include <stdio.h>
#include <pthread.h>
typedef struct {
    float cgpa;
    char* name;
} ThreadData;

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;
```

```
afiamaham@DESKTOP-19OQPQH:~/Operating System$ gcc Lab-5/lab5-task03-cgpa.c -o Lab-5/lab5-task03-cgpa-out
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task03-cgpa-out
Thread 3.640000 says: Afia
All threads done.
afiamaham@DESKTOP-19OQPQH:~/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;

   }
```



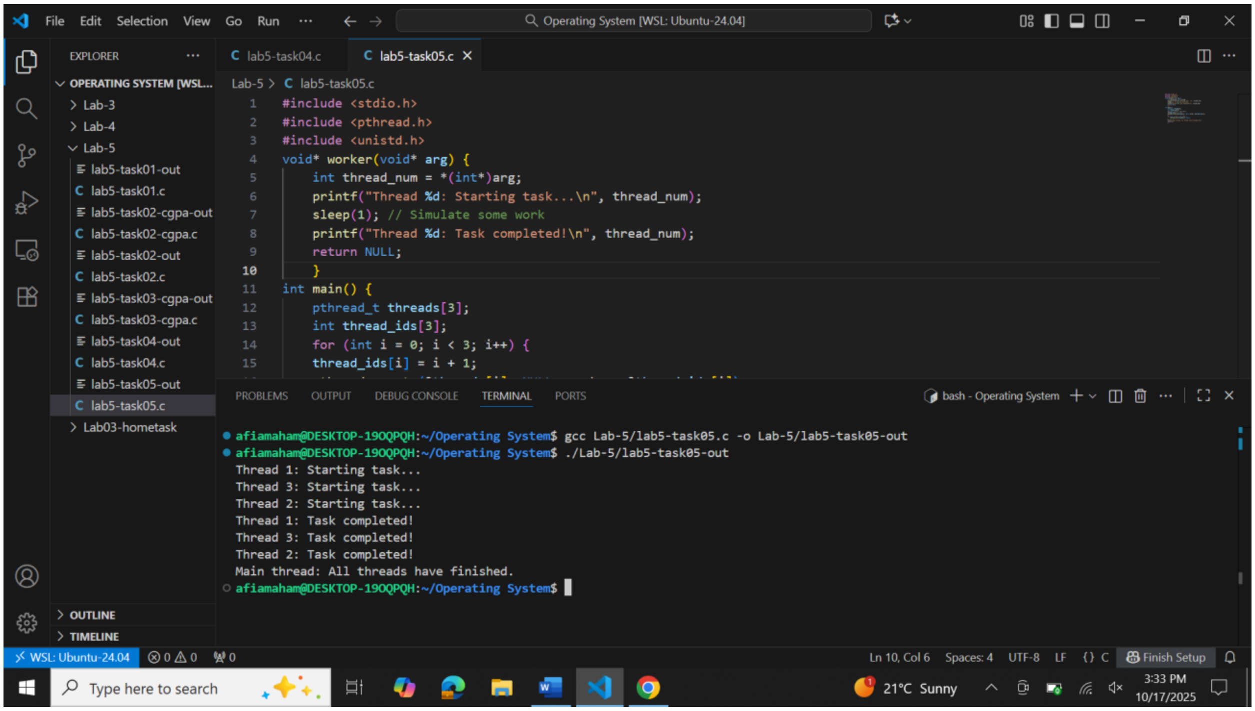
```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
    int n = *(int*)arg;
    int* result = malloc(sizeof(int)); // Allocate memory for result
    *result = 0;
    for (int i = 1; i <= n; i++) {
    *result += i;
    }
    printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
    return (void*)result; // Return the result
    }
int main() {
```

```
afiamaham@DESKTOP-19OQPQH:~/Operating System$ gcc Lab-5/lab5-task04.c -o Lab-5/lab5-task04-out
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task04-out
Thread calculated sum of 1 to 100 = 5050
Main received result: 5050
afiamaham@DESKTOP-19OQPQH:~/Operating System$
```

Task 05:

```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;

}
```

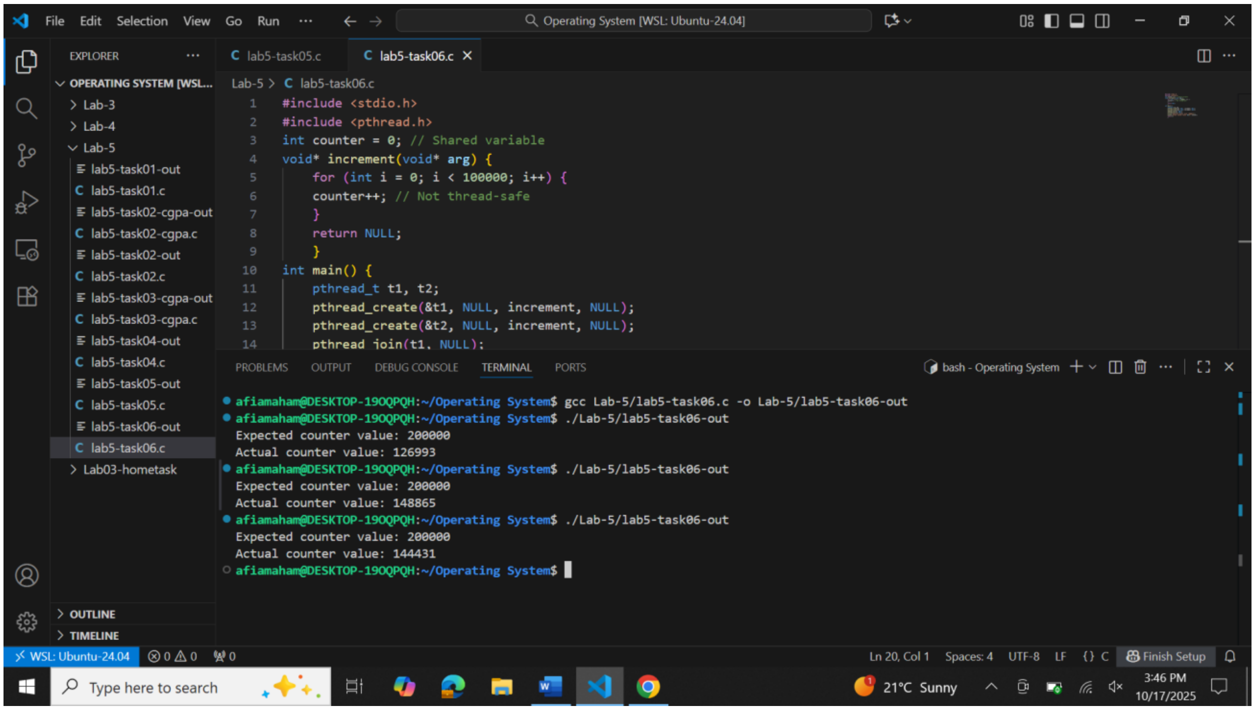


```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* worker(void* arg) {
    int thread_num = *(int*)arg;
    printf("Thread %d: Starting task...\n", thread_num);
    sleep(1); // Simulate some work
    printf("Thread %d: Task completed!\n", thread_num);
    return NULL;
    }
int main() {
    pthread_t threads[3];
    int thread_ids[3];
    for (int i = 0; i < 3; i++) {
    thread_ids[i] = i + 1;
```

```
afiamaham@DESKTOP-19OQPQH:~/Operating System$ gcc Lab-5/lab5-task05.c -o Lab-5/lab5-task05-out
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task05-out
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.
afiamaham@DESKTOP-19OQPQH:~/Operating System$
```

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;

    }
```



```c
#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);
```

```
afiamaham@DESKTOP-19OQPQH:~/Operating System$ gcc Lab-5/lab5-task06.c -o Lab-5/lab5-task06-out
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 126993
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 148865
afiamaham@DESKTOP-19OQPQH:~/Operating System$ ./Lab-5/lab5-task06-out
Expected counter value: 200000
Actual counter value: 144431
afiamaham@DESKTOP-19OQPQH:~/Operating System$
```