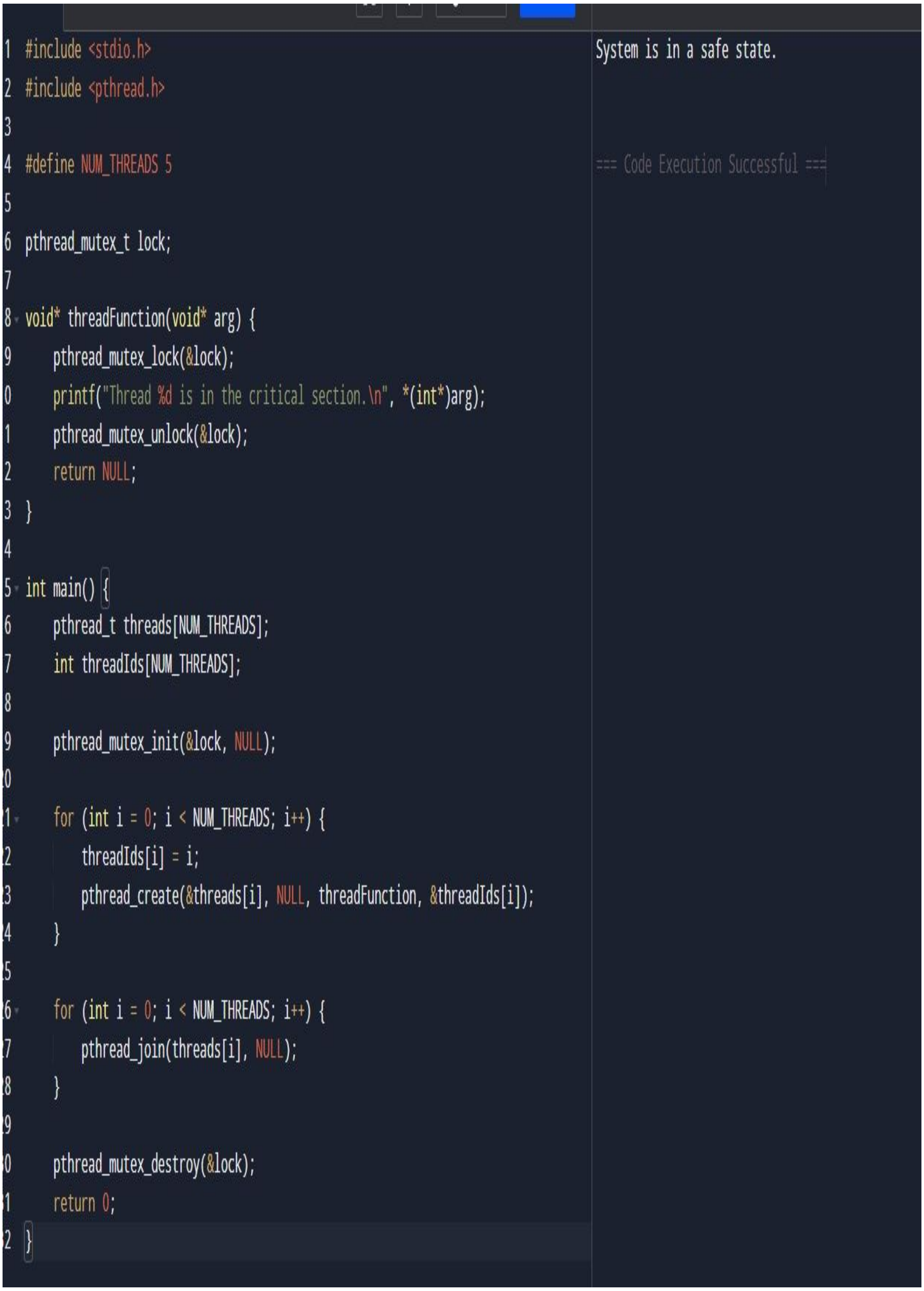
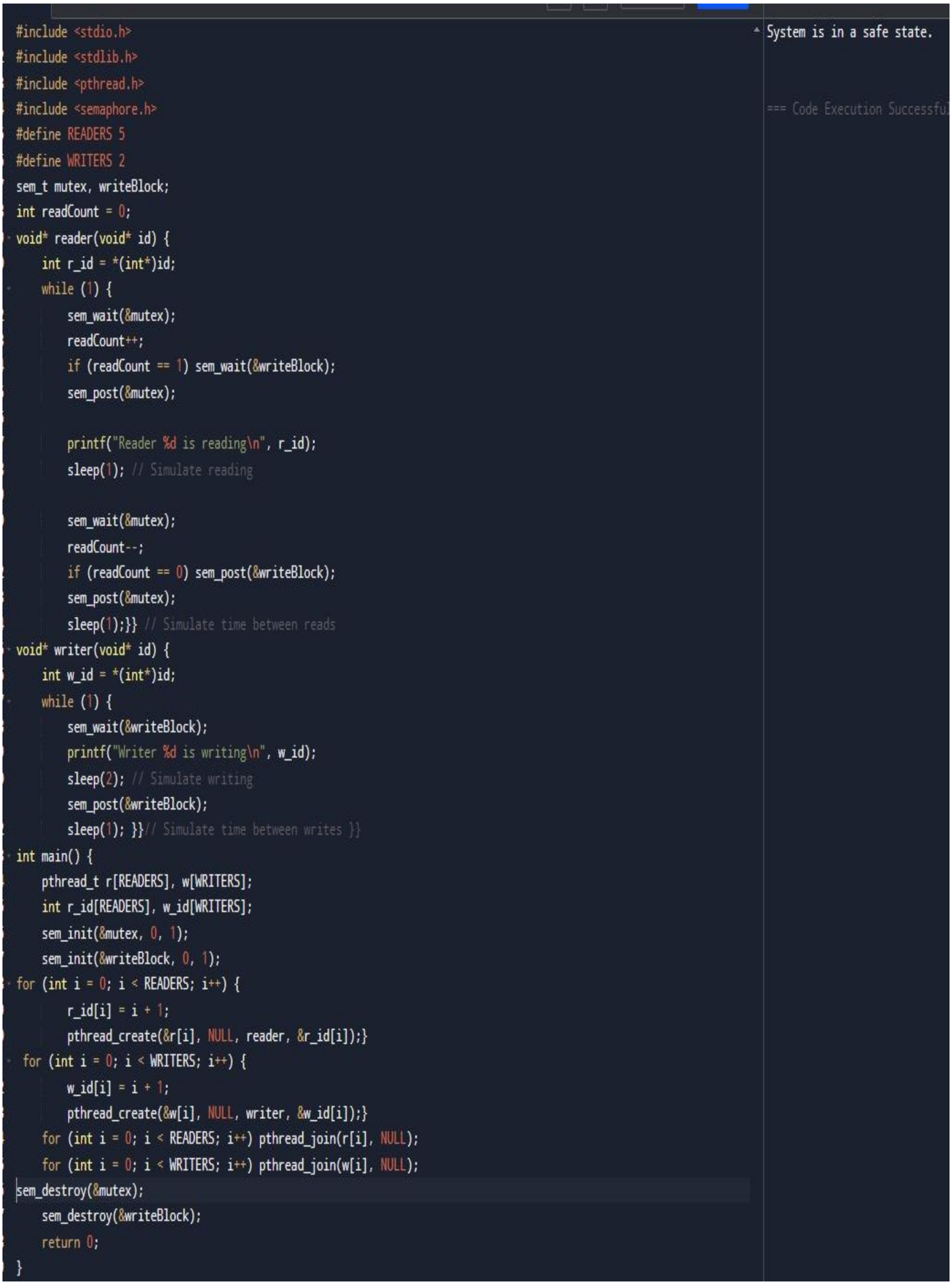


```c
#include <stdio.h>
#include <pthread.h>

#define NUM_THREADS 5

pthread_mutex_t lock;

void* threadFunction(void* arg) {
    pthread_mutex_lock(&lock);
    printf("Thread %d is in the critical section.\n", *(int*)arg);
    pthread_mutex_unlock(&lock);
    return NULL;
}

int main() {
    pthread_t threads[NUM_THREADS];
    int threadIds[NUM_THREADS];

    pthread_mutex_init(&lock, NULL);

    for (int i = 0; i < NUM_THREADS; i++) {
        threadIds[i] = i;
        pthread_create(&threads[i], NULL, threadFunction, &threadIds[i]);
    }

    for (int i = 0; i < NUM_THREADS; i++) {
        pthread_join(threads[i], NULL);
    }

    pthread_mutex_destroy(&lock);
    return 0;
}
```

```
System is in a safe state.


=== Code Execution Successful ===
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#define READERS 5
#define WRITERS 2
sem_t mutex, writeBlock;
int readCount = 0;
void* reader(void* id) {
    int r_id = *(int*)id;
    while (1) {
        sem_wait(&mutex);
        readCount++;
        if (readCount == 1) sem_wait(&writeBlock);
        sem_post(&mutex);

        printf("Reader %d is reading\n", r_id);
        sleep(1); // Simulate reading

        sem_wait(&mutex);
        readCount--;
        if (readCount == 0) sem_post(&writeBlock);
        sem_post(&mutex);
        sleep(1);}} // Simulate time between reads
void* writer(void* id) {
    int w_id = *(int*)id;
    while (1) {
        sem_wait(&writeBlock);
        printf("Writer %d is writing\n", w_id);
        sleep(2); // Simulate writing
        sem_post(&writeBlock);
        sleep(1); }}// Simulate time between writes }}
int main() {
    pthread_t r[READERS], w[WRITERS];
    int r_id[READERS], w_id[WRITERS];
    sem_init(&mutex, 0, 1);
    sem_init(&writeBlock, 0, 1);
for (int i = 0; i < READERS; i++) {
        r_id[i] = i + 1;
        pthread_create(&r[i], NULL, reader, &r_id[i]);}
 for (int i = 0; i < WRITERS; i++) {
        w_id[i] = i + 1;
        pthread_create(&w[i], NULL, writer, &w_id[i]);}
    for (int i = 0; i < READERS; i++) pthread_join(r[i], NULL);
    for (int i = 0; i < WRITERS; i++) pthread_join(w[i], NULL);
sem_destroy(&mutex);
    sem_destroy(&writeBlock);
    return 0;
}
```

```
System is in a safe state.


=== Code Execution Successful
```