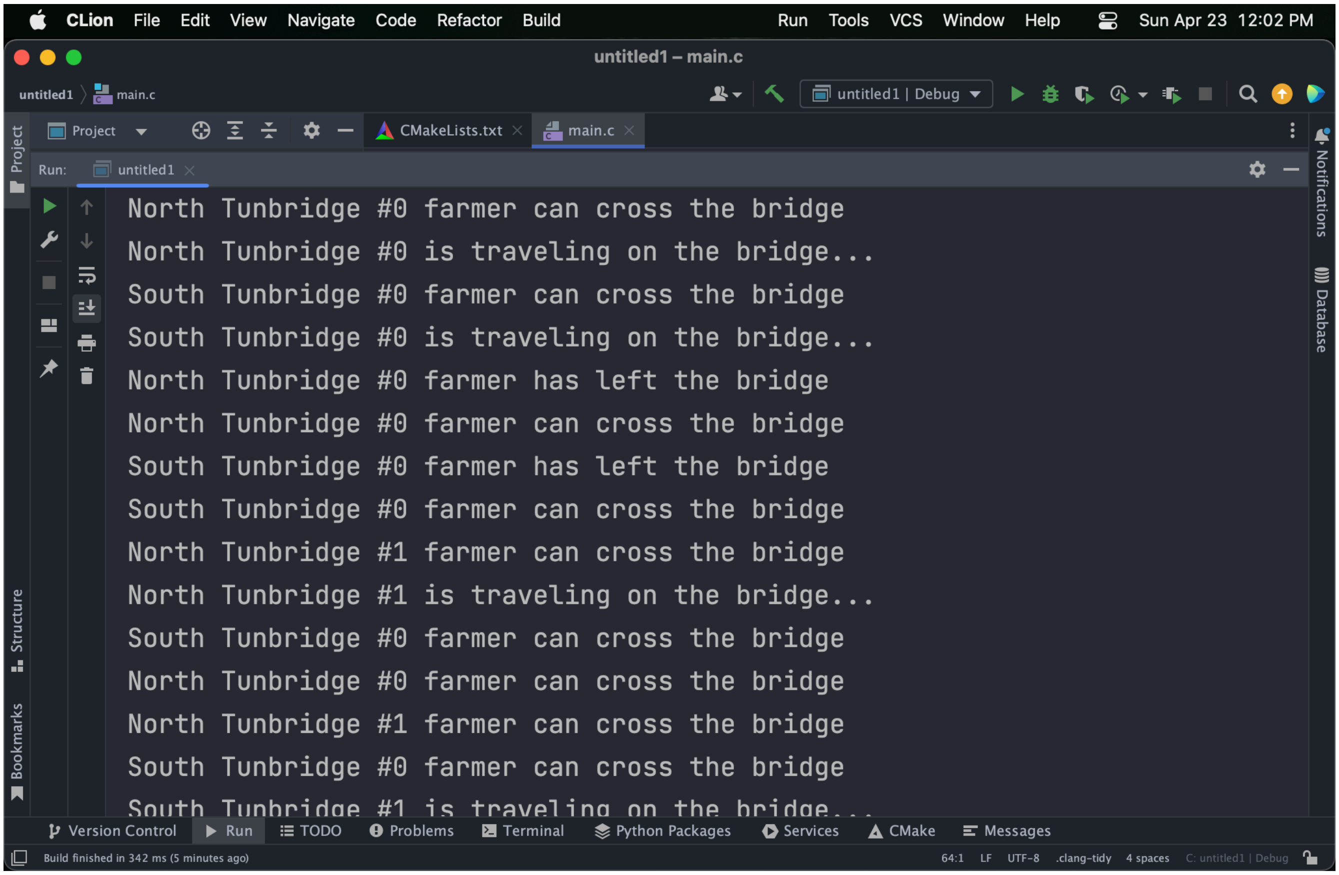
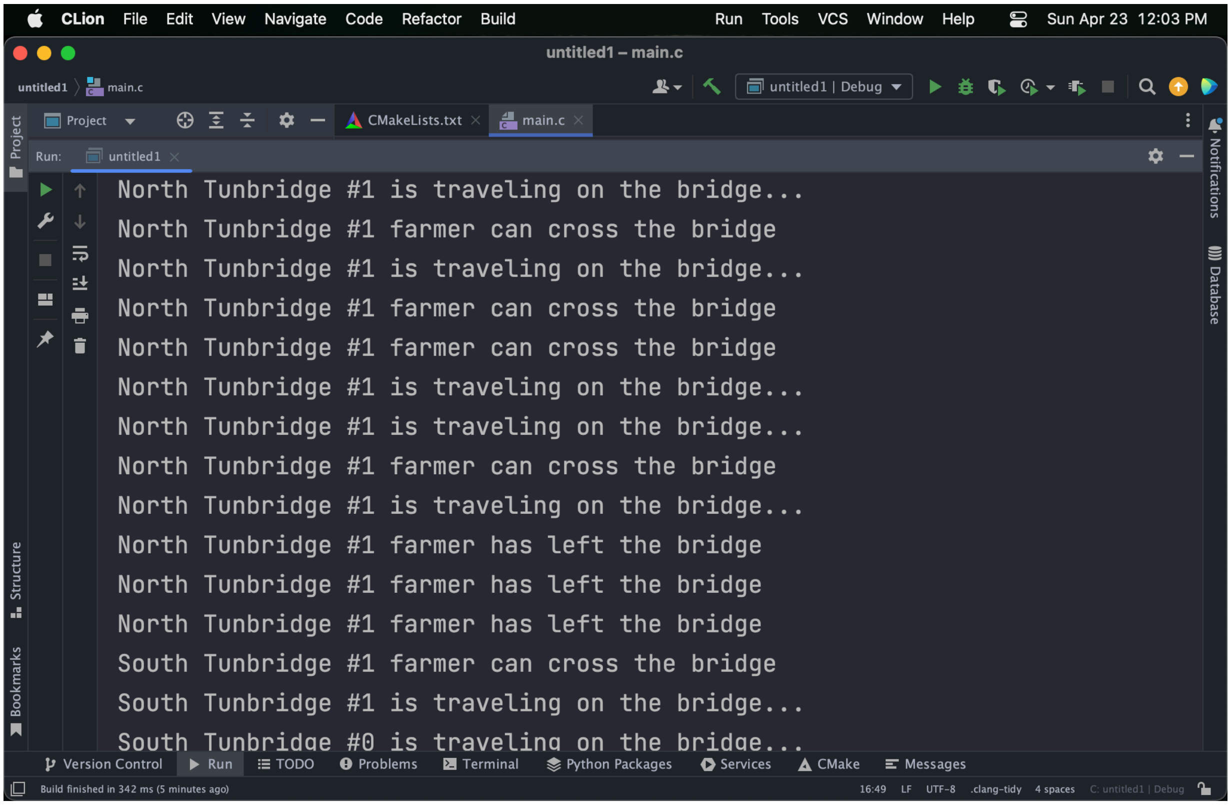
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

#define THREADS 10 // Total Number of Threads

void Bridge(char arr[], int value);

int NorthValue = 0; // Global Variable For North Number
int SouthValue = 0; // Global Variable For South Number
pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER; // Setting Up MUTEX

void *NorthFarmer(void* arg){ // North Farmer Thread
    pthread_mutex_lock(&mutex1); // Critical Section
    char arr[5] = "North"; // North
    printf("%s Tunbridge #%d farmer can cross the bridge\n", arr, NorthValue);
    Bridge(arr,NorthValue); // Enters Bridge
    printf("%s Tunbridge #%d farmer has left the bridge\n", arr, NorthValue);
    NorthValue++; // Increase North Number
    pthread_mutex_unlock(&mutex1); // End of Critical Section
    pthread_exit(0); // Exit Thread.
}

void *SouthFarmer(void* arg){
    pthread_mutex_lock(&mutex1); // Critical Section
    char arr[5] = "South"; // South
    printf("%s Tunbridge #%d farmer can cross the bridge\n", arr, SouthValue);
    Bridge(arr,SouthValue); // Enters Bridge
    printf("%s Tunbridge #%d farmer has left the bridge\n", arr, SouthValue);
    SouthValue++; // Increase South Number
    pthread_mutex_unlock(&mutex1); // End of Critical Section
    pthread_exit(0); // Exit Thread.
}

void Bridge(char arr[], int value){ // Read File Function
    srand(time(NULL)); // Different Seed For Random Number Generator
    printf("%s Tunbridge #%d is traveling on the bridge...\n", arr, value);
    int randomnumber = rand() % 4; // Gets Random Number From 0 to 3
    sleep(randomnumber); // Sleep For up to 3 Seconds
    //printf("Random %d \n", randomnumber);
}


int main(){
    pthread_t North[THREADS]; // North Thread
    pthread_t South[THREADS]; // South Thread
    pthread_mutex_init(&mutex1,NULL); // Initializing Mutex

    for(int i = 0; i < THREADS; i++){ // Joining of The Threads
        int CreateFirst =  pthread_create(&North[i], NULL, NorthFarmer, NULL); // Thread For North
        int CreateSecond = pthread_create(&South[i], NULL, SouthFarmer, NULL); // Thread For South
        if(CreateFirst != 0 || CreateSecond != 0){ // In Case Thread Create Fails
            fprintf(stderr, "Thread Create Failed");
            return 1;
        }
    }

    for(int i = 0; i < THREADS; i++){ // Joining of The Threads
```

```
        int JoinFirst = pthread_join(North[i],NULL); // Joining North
        int JoinSecond = pthread_join(South[i],NULL); // Joining South
         if(JoinFirst != 0 || JoinSecond != 0){ // In Case Thread Join Fails
            fprintf(stderr, "Join Failed");
            return 1;
        }
    }
    pthread_mutex_destroy(&mutex1); // Destroying Mutex
    return 0;
}
```

No Lock



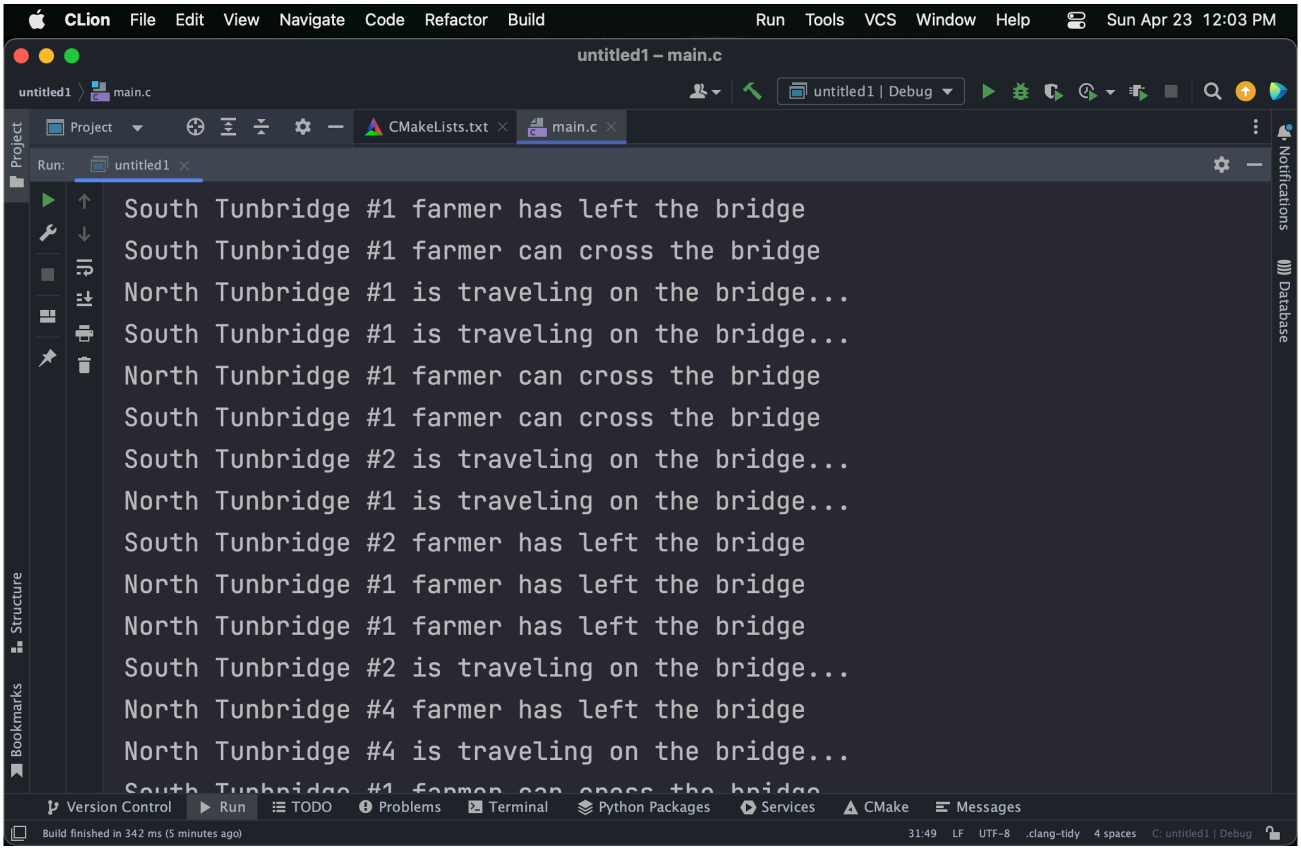
```
North Tunbridge #0 farmer can cross the bridge
North Tunbridge #0 is traveling on the bridge...
South Tunbridge #0 farmer can cross the bridge
South Tunbridge #0 is traveling on the bridge...
North Tunbridge #0 farmer has left the bridge
North Tunbridge #0 farmer can cross the bridge
South Tunbridge #0 farmer has left the bridge
South Tunbridge #0 farmer can cross the bridge
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
South Tunbridge #0 farmer can cross the bridge
North Tunbridge #0 farmer can cross the bridge
North Tunbridge #1 farmer can cross the bridge
South Tunbridge #0 farmer can cross the bridge
South Tunbridge #1 is traveling on the bridge...
```



CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Sun Apr 23 12:03 PM
untitled1 – main.c
untitled1 main.c
untitled1 | Debug
Project
CMakeLists.txt
main.c
Run: untitled1
Notifications
Database
Structure
Bookmarks
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
16:49 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug

```
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer has left the bridge
North Tunbridge #1 farmer has left the bridge
North Tunbridge #1 farmer has left the bridge
South Tunbridge #1 farmer can cross the bridge
South Tunbridge #1 is traveling on the bridge...
South Tunbridge #0 is traveling on the bridge...
```

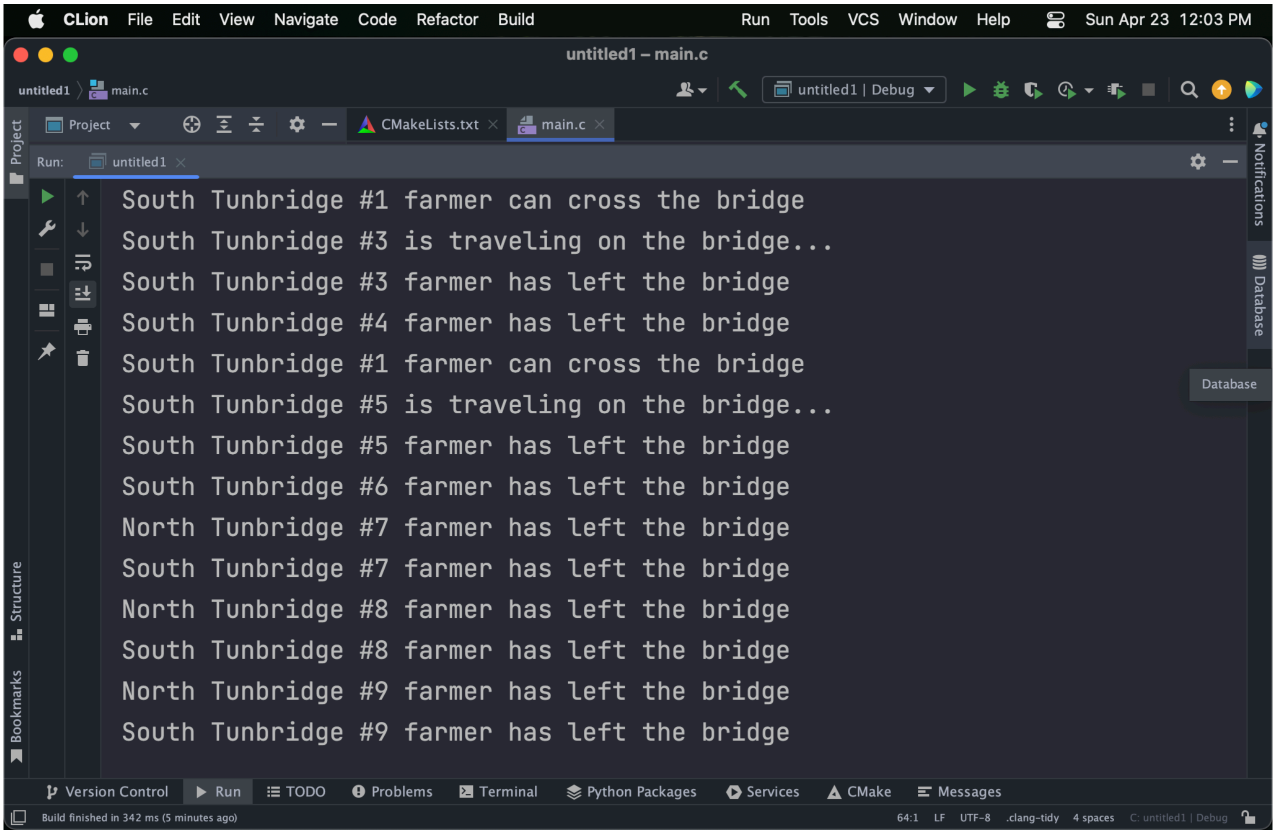
Build finished in 342 ms (5 minutes ago)



CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Sun Apr 23 12:03 PM
untitled1 – main.c
untitled1 main.c
untitled1 | Debug
Project
CMakeLists.txt
main.c
Run: untitled1
Notifications
Database
Structure
Bookmarks
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
31:49 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug

```
South Tunbridge #1 farmer has left the bridge
South Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
South Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer can cross the bridge
South Tunbridge #1 farmer can cross the bridge
South Tunbridge #2 is traveling on the bridge...
North Tunbridge #1 is traveling on the bridge...
South Tunbridge #2 farmer has left the bridge
North Tunbridge #1 farmer has left the bridge
North Tunbridge #1 farmer has left the bridge
South Tunbridge #2 is traveling on the bridge...
North Tunbridge #4 farmer has left the bridge
North Tunbridge #4 is traveling on the bridge...
South Tunbridge #1 farmer can cross the bridge
```

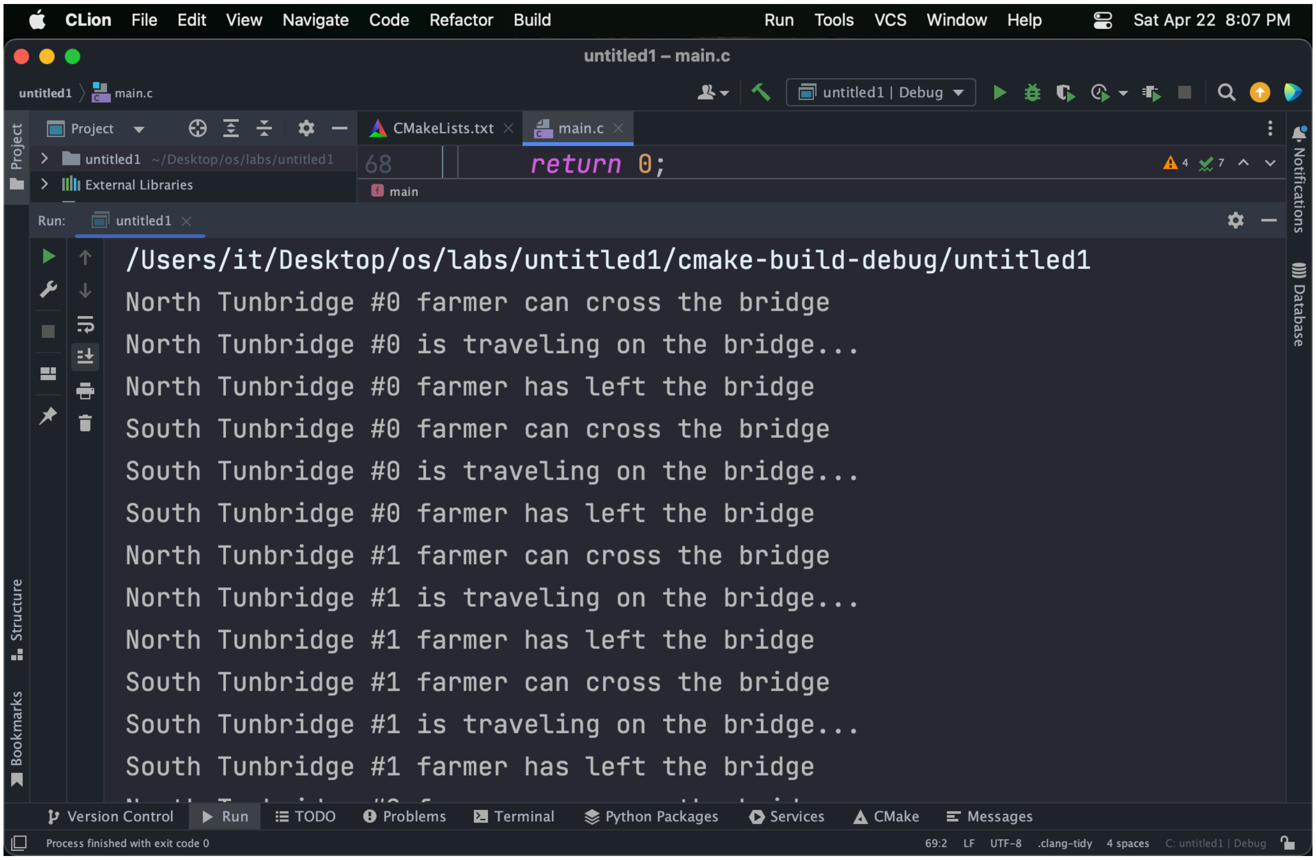
Build finished in 342 ms (5 minutes ago)



```
South Tunbridge #1 farmer can cross the bridge
South Tunbridge #3 is traveling on the bridge...
South Tunbridge #3 farmer has left the bridge
South Tunbridge #4 farmer has left the bridge
South Tunbridge #1 farmer can cross the bridge
South Tunbridge #5 is traveling on the bridge...
South Tunbridge #5 farmer has left the bridge
South Tunbridge #6 farmer has left the bridge
North Tunbridge #7 farmer has left the bridge
South Tunbridge #7 farmer has left the bridge
North Tunbridge #8 farmer has left the bridge
South Tunbridge #8 farmer has left the bridge
North Tunbridge #9 farmer has left the bridge
South Tunbridge #9 farmer has left the bridge
```

Build finished in 342 ms (5 minutes ago)

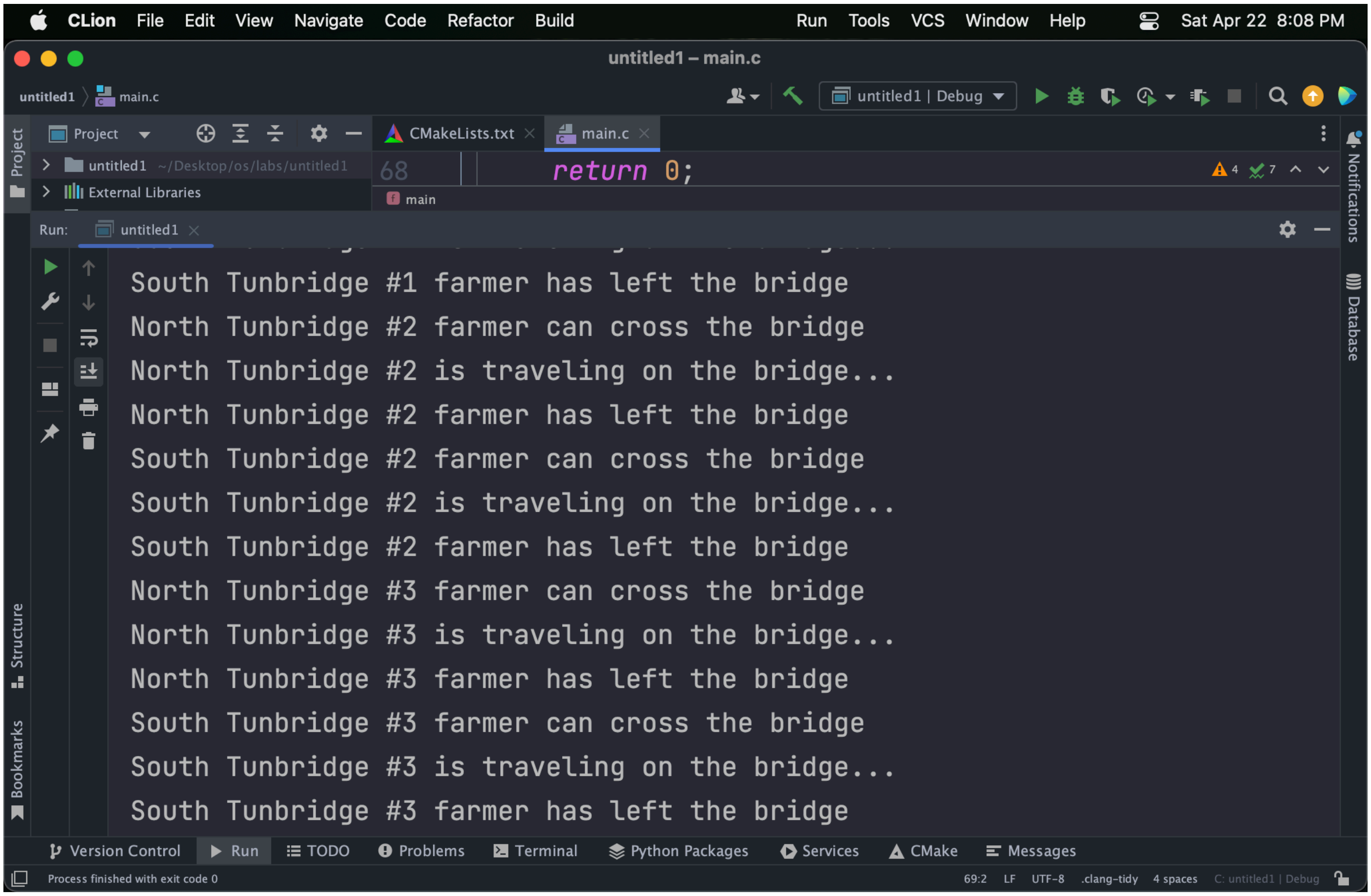
With Lock



```
    return 0;
```

```
/Users/it/Desktop/os/labs/untitled1/cmake-build-debug/untitled1
North Tunbridge #0 farmer can cross the bridge
North Tunbridge #0 is traveling on the bridge...
North Tunbridge #0 farmer has left the bridge
South Tunbridge #0 farmer can cross the bridge
South Tunbridge #0 is traveling on the bridge...
South Tunbridge #0 farmer has left the bridge
North Tunbridge #1 farmer can cross the bridge
North Tunbridge #1 is traveling on the bridge...
North Tunbridge #1 farmer has left the bridge
South Tunbridge #1 farmer can cross the bridge
South Tunbridge #1 is traveling on the bridge...
South Tunbridge #1 farmer has left the bridge
```

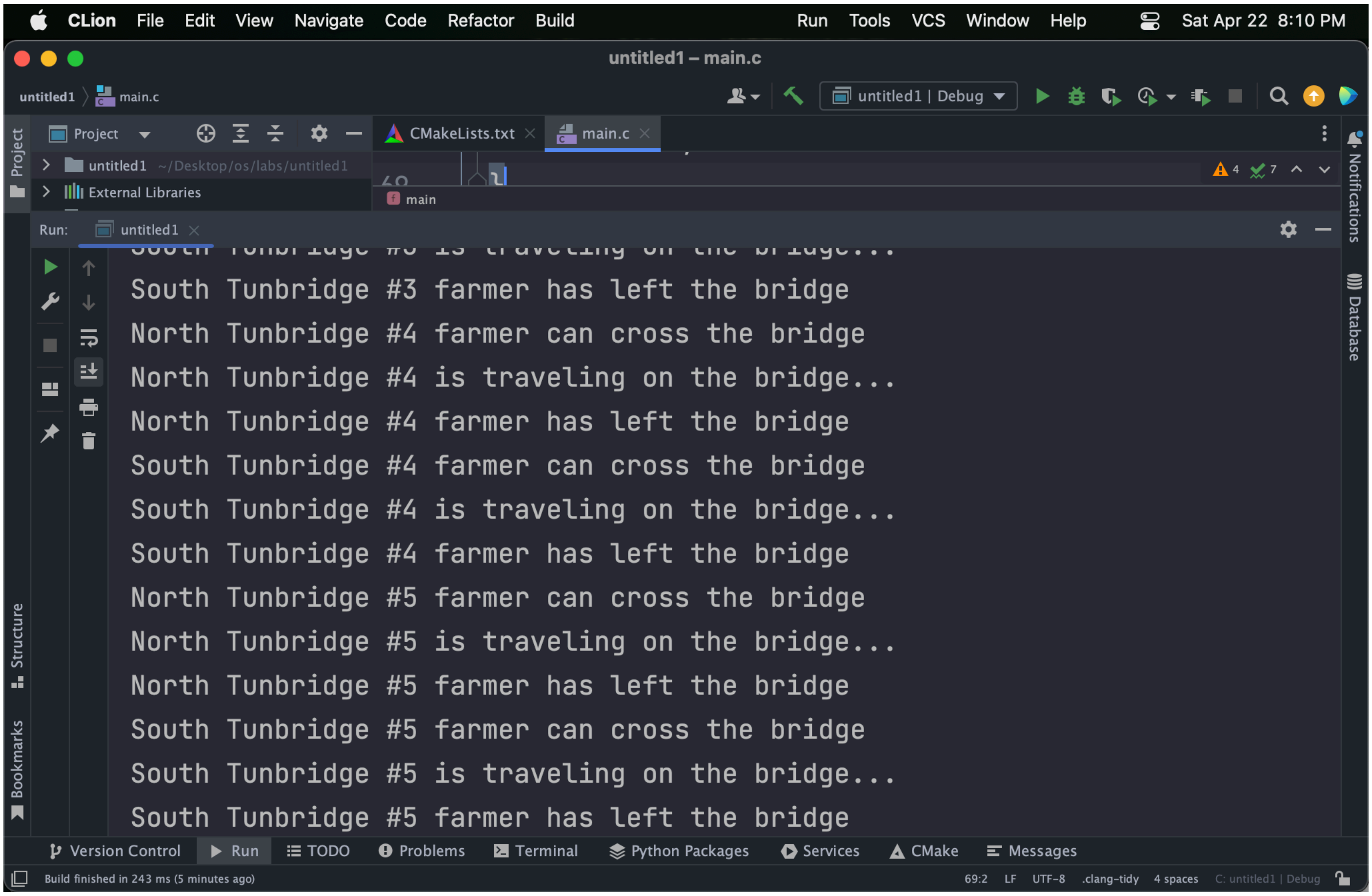
Process finished with exit code 0


CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Sat Apr 22 8:08 PM
untitled1 – main.c
untitled1 | Debug
Project
untitled1 ~/Desktop/os/labs/untitled1
External Libraries
CMakeLists.txt
main.c
main
Run: untitled1
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
69:2 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug

```
return 0;
```

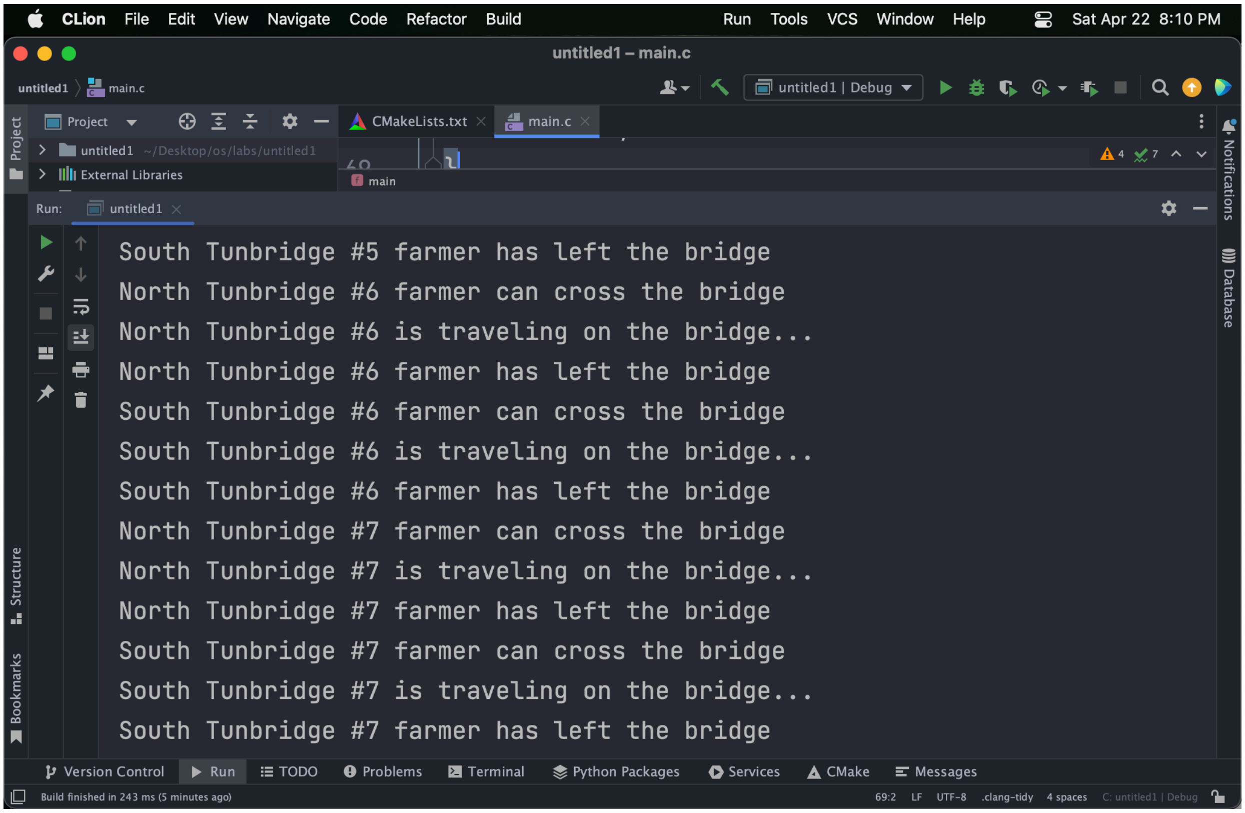
```
South Tunbridge #1 farmer has left the bridge
North Tunbridge #2 farmer can cross the bridge
North Tunbridge #2 is traveling on the bridge...
North Tunbridge #2 farmer has left the bridge
South Tunbridge #2 farmer can cross the bridge
South Tunbridge #2 is traveling on the bridge...
South Tunbridge #2 farmer has left the bridge
North Tunbridge #3 farmer can cross the bridge
North Tunbridge #3 is traveling on the bridge...
North Tunbridge #3 farmer has left the bridge
South Tunbridge #3 farmer can cross the bridge
South Tunbridge #3 is traveling on the bridge...
South Tunbridge #3 farmer has left the bridge
```

Process finished with exit code 0


CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Sat Apr 22 8:10 PM
untitled1 – main.c
untitled1 | Debug
Project
untitled1 ~/Desktop/os/labs/untitled1
External Libraries
CMakeLists.txt
main.c
main
Run: untitled1
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
69:2 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug

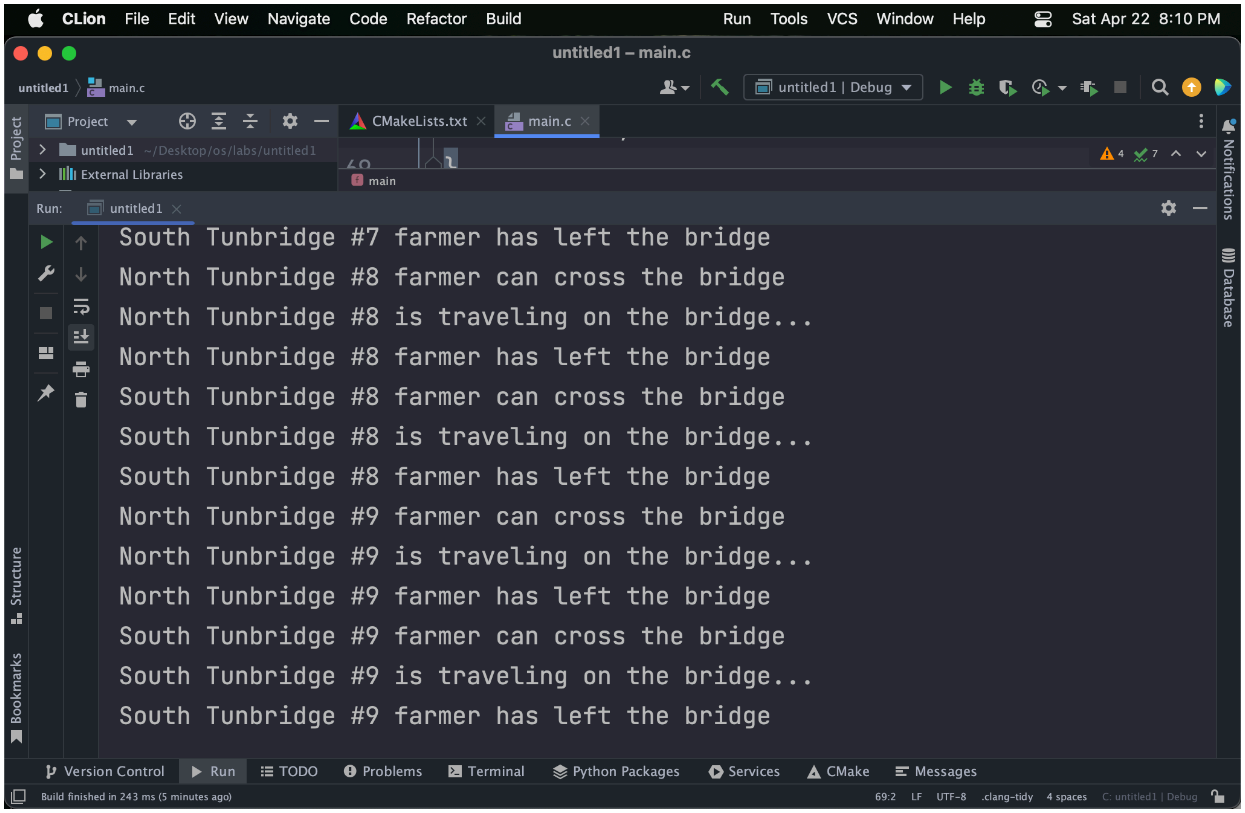
```
South Tunbridge #3 farmer has left the bridge
North Tunbridge #4 farmer can cross the bridge
North Tunbridge #4 is traveling on the bridge...
North Tunbridge #4 farmer has left the bridge
South Tunbridge #4 farmer can cross the bridge
South Tunbridge #4 is traveling on the bridge...
South Tunbridge #4 farmer has left the bridge
North Tunbridge #5 farmer can cross the bridge
North Tunbridge #5 is traveling on the bridge...
North Tunbridge #5 farmer has left the bridge
South Tunbridge #5 farmer can cross the bridge
South Tunbridge #5 is traveling on the bridge...
South Tunbridge #5 farmer has left the bridge
```

Build finished in 243 ms (5 minutes ago)



CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Sat Apr 22 8:10 PM
untitled1 – main.c
untitled1 main.c
untitled1 | Debug
Project
CMakeLists.txt
main.c
untitled1 ~/Desktop/os/labs/untitled1
External Libraries
main
Run: untitled1
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
Build finished in 243 ms (5 minutes ago)
69:2 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug
Notifications
Database
Structure
Bookmarks

```
South Tunbridge #5 farmer has left the bridge
North Tunbridge #6 farmer can cross the bridge
North Tunbridge #6 is traveling on the bridge...
North Tunbridge #6 farmer has left the bridge
South Tunbridge #6 farmer can cross the bridge
South Tunbridge #6 is traveling on the bridge...
South Tunbridge #6 farmer has left the bridge
North Tunbridge #7 farmer can cross the bridge
North Tunbridge #7 is traveling on the bridge...
North Tunbridge #7 farmer has left the bridge
South Tunbridge #7 farmer can cross the bridge
South Tunbridge #7 is traveling on the bridge...
South Tunbridge #7 farmer has left the bridge
```



CLion File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Sat Apr 22 8:10 PM
untitled1 – main.c
untitled1 main.c
untitled1 | Debug
Project
CMakeLists.txt
main.c
untitled1 ~/Desktop/os/labs/untitled1
External Libraries
main
Run: untitled1
Version Control Run TODO Problems Terminal Python Packages Services CMake Messages
Build finished in 243 ms (5 minutes ago)
69:2 LF UTF-8 .clang-tidy 4 spaces C: untitled1 | Debug
Notifications
Database
Structure
Bookmarks

```
South Tunbridge #7 farmer has left the bridge
North Tunbridge #8 farmer can cross the bridge
North Tunbridge #8 is traveling on the bridge...
North Tunbridge #8 farmer has left the bridge
South Tunbridge #8 farmer can cross the bridge
South Tunbridge #8 is traveling on the bridge...
South Tunbridge #8 farmer has left the bridge
North Tunbridge #9 farmer can cross the bridge
North Tunbridge #9 is traveling on the bridge...
North Tunbridge #9 farmer has left the bridge
South Tunbridge #9 farmer can cross the bridge
South Tunbridge #9 is traveling on the bridge...
South Tunbridge #9 farmer has left the bridge
```

What type of lock your code is using?

Mutex lock is like a lock around section of code. First we have to initialized with pthread_mutex_init(&mutex, NULL) and this one comes with destroying allocated memory pthread_mutex_destroy(&mutex). Now mutex is available, so we can use it with pthread_mutex_lock(&mutex) and after the code unlock it pthread_mutex_unlock(&mutex). They like protection, that other threads won't execute it at a same time. So whatever code located between lock and unlock won't be executed by other threads. That is why name lock, because it is like lock, we locked it and nobody can access it until its been unlock. If wouldn't have mutex lock we would get into race condition (as screenshots shows that will be chaotic execution by threads without mutex lock).

How many processes does your program fork? It is 60 processes.
Are there equal number of North and South processes? Yes, there are 30 processes for North and 30 processes for South.