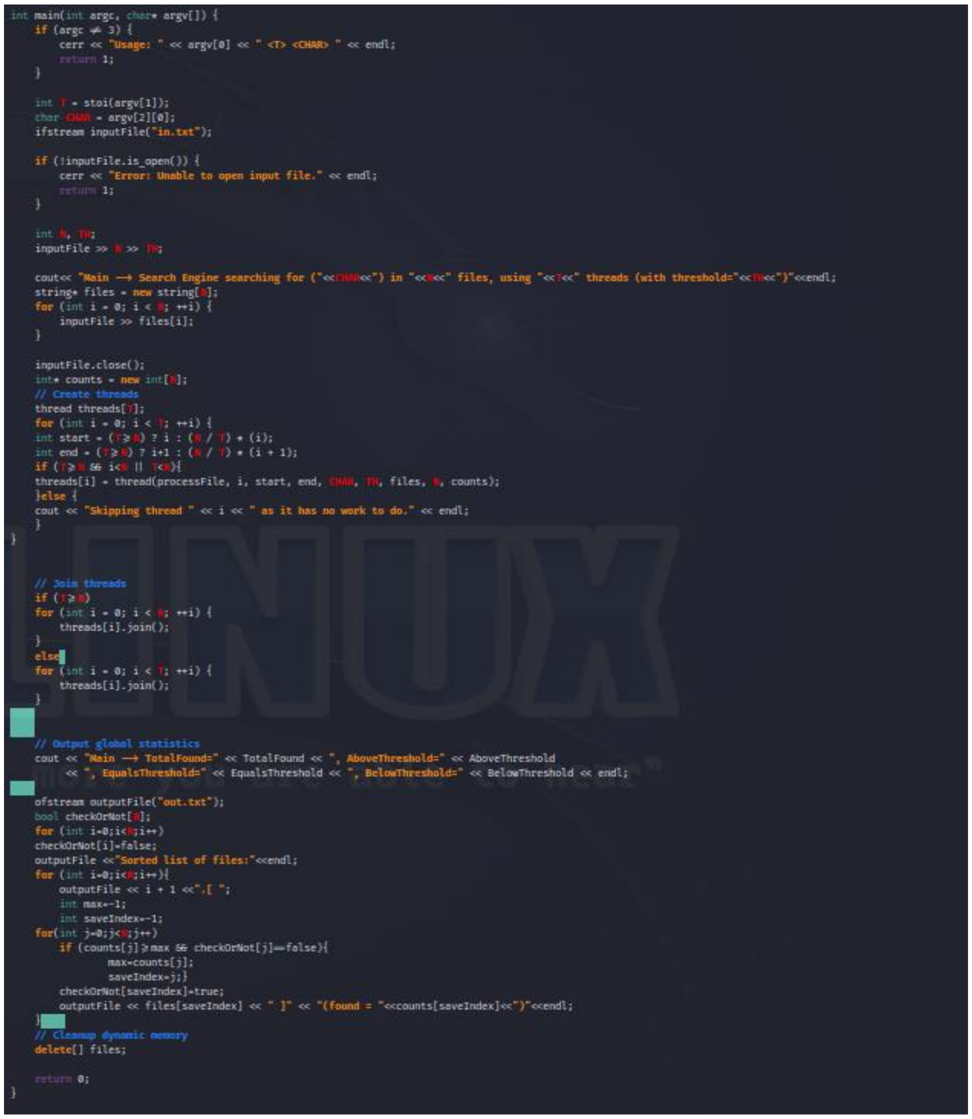
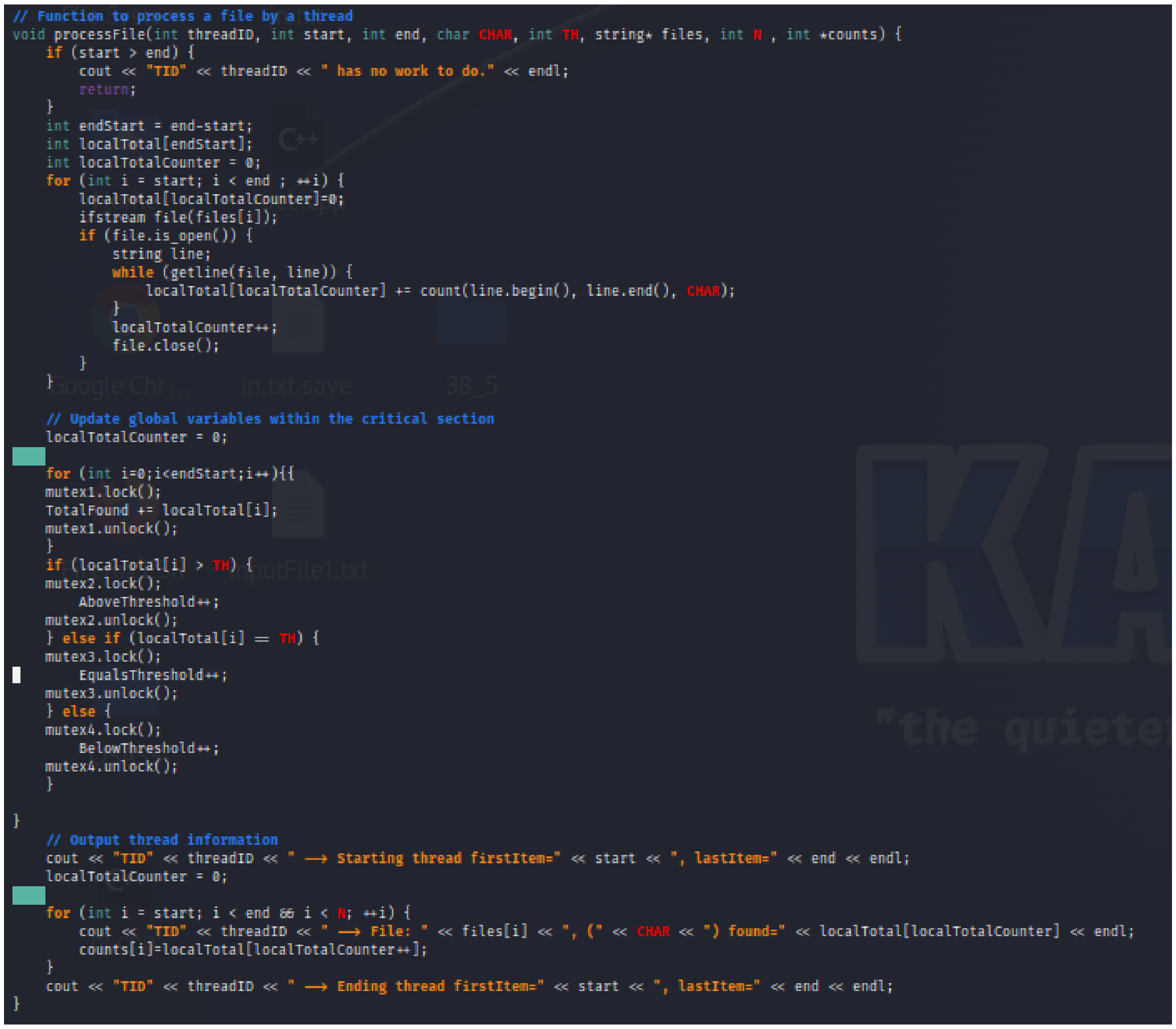
Main ThreadSafe-NonThreadSafe:



```
int main(int argc, char* argv[]) {
    if (argc ≠ 3) {
        cerr << "Usage: " << argv[0] << " <T> <CHAR> " << endl;
        return 1;
    }

    int T = stoi(argv[1]);
    char CHAR = argv[2][0];
    ifstream inputFile("in.txt");

    if (!inputFile.is_open()) {
        cerr << "Error: Unable to open input file." << endl;
        return 1;
    }

    int N, TH;
    inputFile >> N >> TH;

    cout<< "Main ⟶ Search Engine searching for ("<<CHAR<<") in "<<N<<" files, using "<<T<<" threads (with threshold="<<TH<<")"<<endl;
    string* files = new string[N];
    for (int i = 0; i < N; ++i) {
        inputFile >> files[i];
    }

    inputFile.close();
    int* counts = new int[N];
    // Create threads
    thread threads[T];
    for (int i = 0; i < T; ++i) {
    int start = (T≥N) ? i : (N / T) * (i);
    int end = (T≥N) ? i+1 : (N / T) * (i + 1);
    if (T≥N && i<N || T<N){
    threads[i] = thread(processFile, i, start, end, CHAR, TH, files, N, counts);
    }else {
    cout << "Skipping thread " << i << " as it has no work to do." << endl;
    }
}


    // Join threads
    if (T≥N)
    for (int i = 0; i < N; ++i) {
        threads[i].join();
    }
    else
    for (int i = 0; i < T; ++i) {
        threads[i].join();
    }


    // Output global statistics
    cout << "Main ⟶ TotalFound=" << TotalFound << ", AboveThreshold=" << AboveThreshold
       << ", EqualsThreshold=" << EqualsThreshold << ", BelowThreshold=" << BelowThreshold << endl;

    ofstream outputFile("out.txt");
    bool checkOrNot[N];
    for (int i=0;i<N;i++)
    checkOrNot[i]=false;
    outputFile <<"Sorted list of files:"<<endl;
    for (int i=0;i<N;i++){
        outputFile << i + 1 <<".[ ";
        int max=-1;
        int saveIndex=-1;
    for(int j=0;j<N;j++)
        if (counts[j]≥max && checkOrNot[j]==false){
                max=counts[j];
                saveIndex=j;}
        checkOrNot[saveIndex]=true;
        outputFile << files[saveIndex] << " ]" << "(found = "<<counts[saveIndex]<<")"<<endl;
    }
    // Cleanup dynamic memory
    delete[] files;

    return 0;
}
```

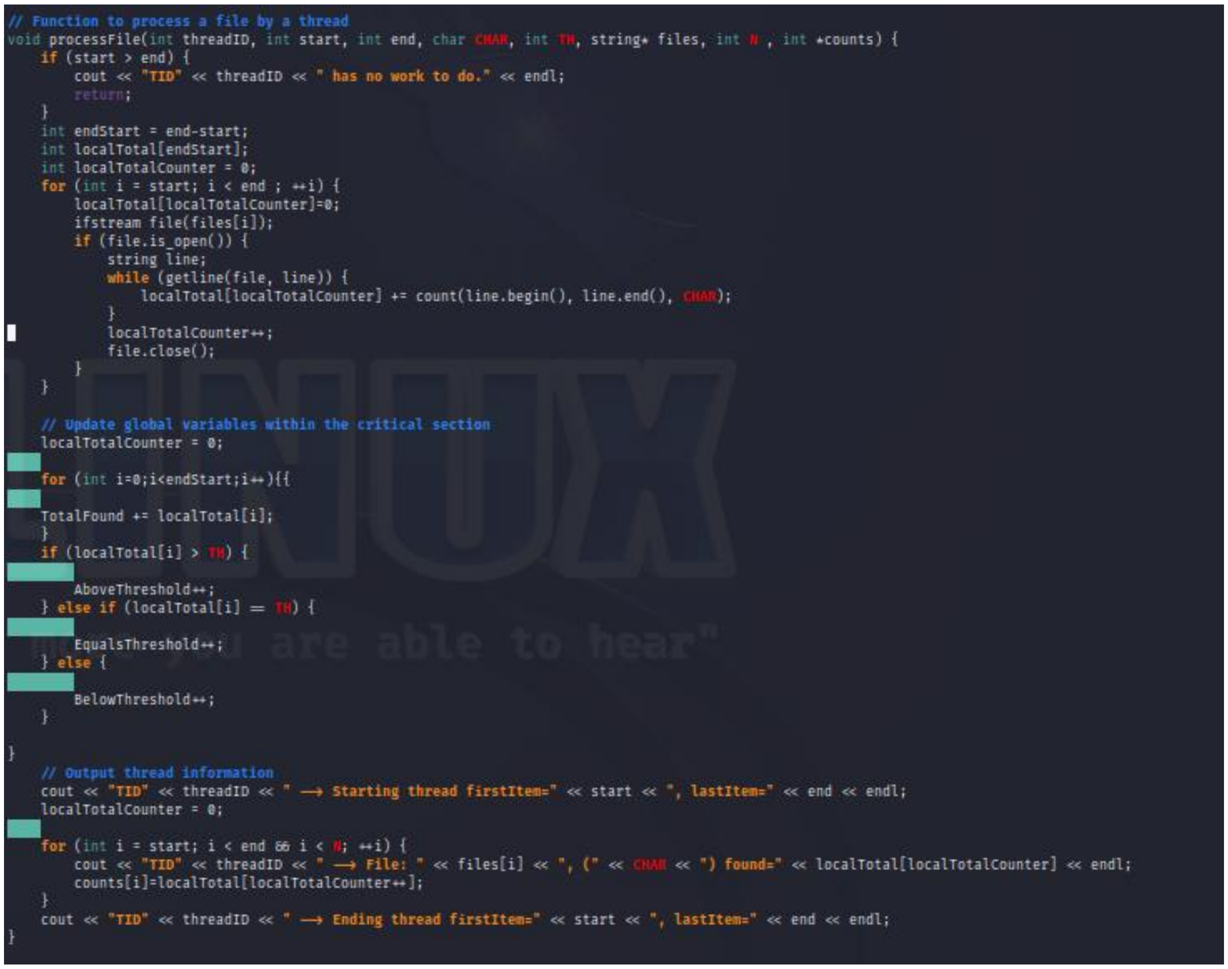
ThreadsafeFunction :



```
// Function to process a file by a thread
void processFile(int threadID, int start, int end, char CHAR, int TH, string* files, int N , int *counts) {
    if (start > end) {
        cout << "TID" << threadID << " has no work to do." << endl;
        return;
    }
    int endStart = end-start;
    int localTotal[endStart];
    int localTotalCounter = 0;
    for (int i = start; i < end ; ++i) {
        localTotal[localTotalCounter]=0;
        ifstream file(files[i]);
        if (file.is_open()) {
            string line;
            while (getline(file, line)) {
                localTotal[localTotalCounter] += count(line.begin(), line.end(), CHAR);
            }
            localTotalCounter++;
            file.close();
        }
    }

    // Update global variables within the critical section
    localTotalCounter = 0;

    for (int i=0;i<endStart;i++){{
    mutex1.lock();
    TotalFound += localTotal[i];
    mutex1.unlock();
    }
    if (localTotal[i] > TH) {
    mutex2.lock();
        AboveThreshold++;
    mutex2.unlock();
    } else if (localTotal[i] == TH) {
    mutex3.lock();
        EqualsThreshold++;
    mutex3.unlock();
    } else {
    mutex4.lock();
        BelowThreshold++;
    mutex4.unlock();
    }

}
    // Output thread information
    cout << "TID" << threadID << " --> Starting thread firstItem=" << start << ", lastItem=" << end << endl;
    localTotalCounter = 0;

    for (int i = start; i < end && i < N; ++i) {
        cout << "TID" << threadID << " --> File: " << files[i] << ", (" << CHAR << ") found=" << localTotal[localTotalCounter] << endl;
        counts[i]=localTotal[localTotalCounter++];
    }
    cout << "TID" << threadID << " --> Ending thread firstItem=" << start << ", lastItem=" << end << endl;
}
```

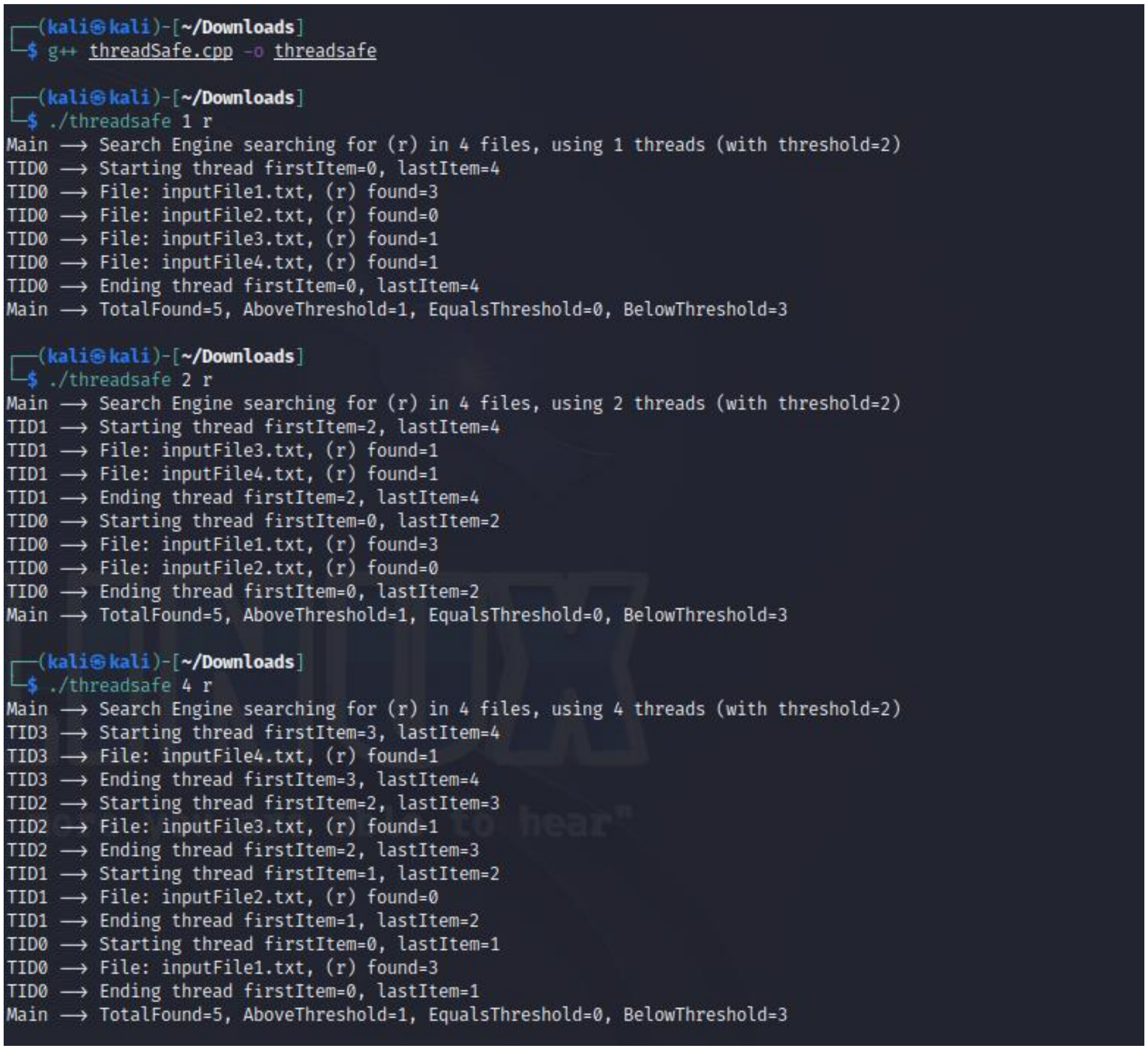
We add Mutex to our Code to prevent the Race condition

ThreadNonsafefunction:
]

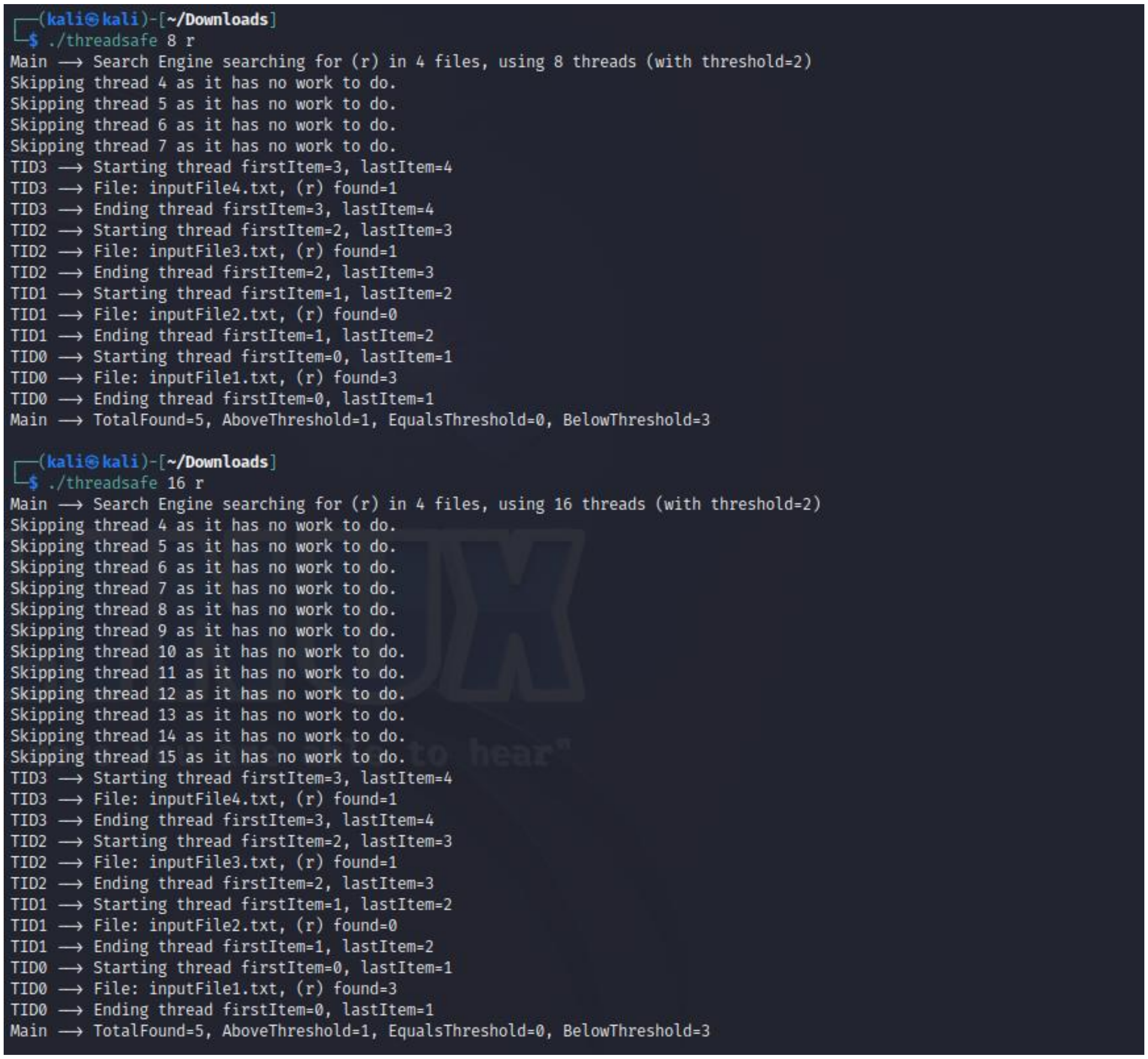


```cpp
// Function to process a file by a thread
void processFile(int threadID, int start, int end, char CHAR, int TH, string* files, int N , int *counts) {
    if (start > end) {
        cout << "TID" << threadID << " has no work to do." << endl;
        return;
    }
    int endStart = end-start;
    int localTotal[endStart];
    int localTotalCounter = 0;
    for (int i = start; i < end ; ++i) {
        localTotal[localTotalCounter]=0;
        ifstream file(files[i]);
        if (file.is_open()) {
            string line;
            while (getline(file, line)) {
                localTotal[localTotalCounter] += count(line.begin(), line.end(), CHAR);
            }
            localTotalCounter++;
            file.close();
        }
    }

    // Update global variables within the critical section
    localTotalCounter = 0;

    for (int i=0;i<endStart;i++){{

    TotalFound += localTotal[i];
    }
    if (localTotal[i] > TH) {

        AboveThreshold++;
    } else if (localTotal[i] == TH) {

        EqualsThreshold++;
    } else {

        BelowThreshold++;
    }

}
    // Output thread information
    cout << "TID" << threadID << " --> Starting thread firstItem=" << start << ", lastItem=" << end << endl;
    localTotalCounter = 0;

    for (int i = start; i < end && i < N; ++i) {
        cout << "TID" << threadID << " --> File: " << files[i] << ", (" << CHAR << ") found=" << localTotal[localTotalCounter] << endl;
        counts[i]=localTotal[localTotalCounter++];
    }
    cout << "TID" << threadID << " --> Ending thread firstItem=" << start << ", lastItem=" << end << endl;
}
```

testcaseThreadSafe:

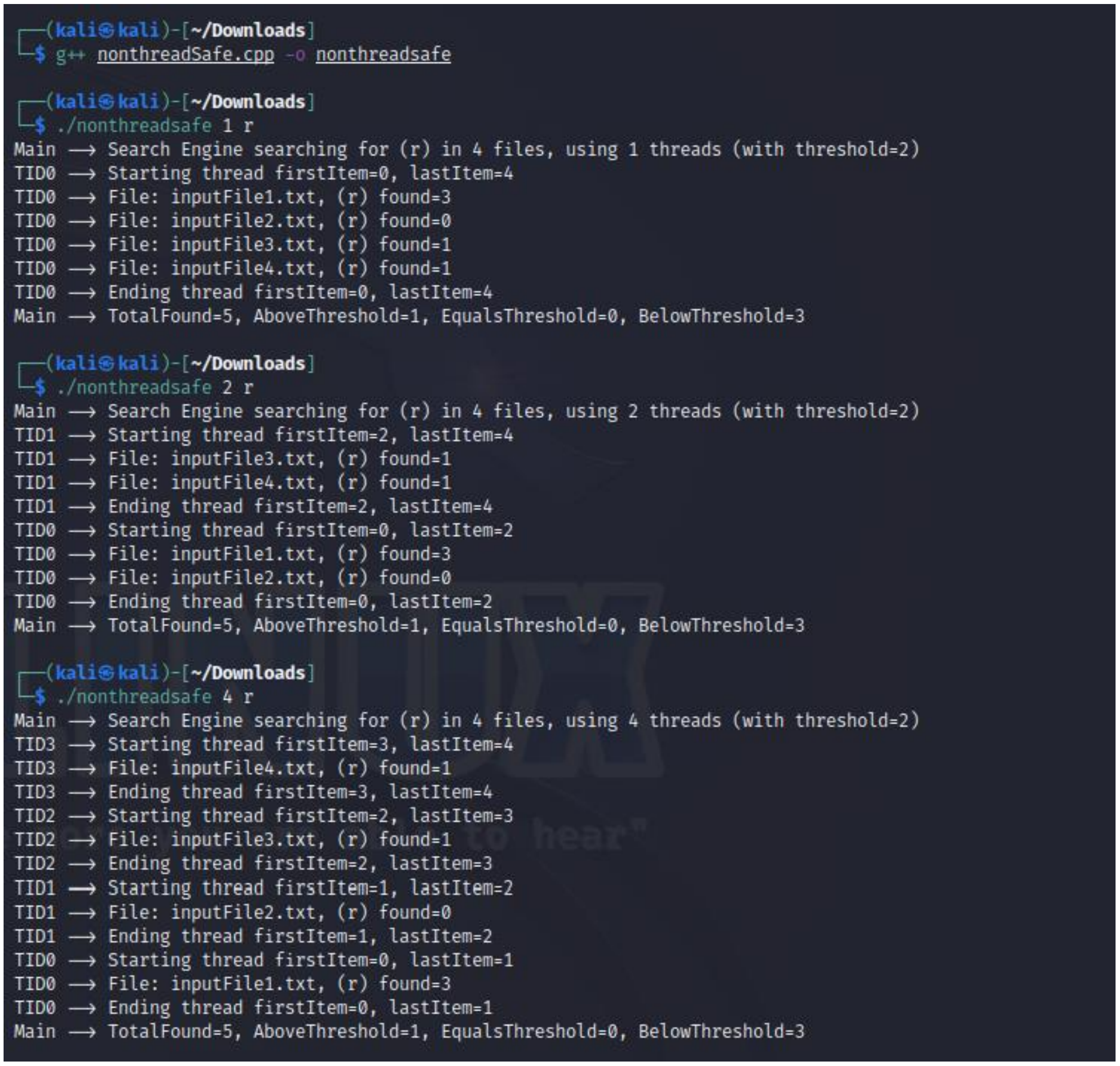


```
┌──(kali㉿kali)-[~/Downloads]
└─$ g++ threadSafe.cpp -o threadsafe

┌──(kali㉿kali)-[~/Downloads]
└─$ ./threadsafe 1 r
Main ⟶ Search Engine searching for (r) in 4 files, using 1 threads (with threshold=2)
TID0 ⟶ Starting thread firstItem=0, lastItem=4
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ File: inputFile2.txt, (r) found=0
TID0 ⟶ File: inputFile3.txt, (r) found=1
TID0 ⟶ File: inputFile4.txt, (r) found=1
TID0 ⟶ Ending thread firstItem=0, lastItem=4
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./threadsafe 2 r
Main ⟶ Search Engine searching for (r) in 4 files, using 2 threads (with threshold=2)
TID1 ⟶ Starting thread firstItem=2, lastItem=4
TID1 ⟶ File: inputFile3.txt, (r) found=1
TID1 ⟶ File: inputFile4.txt, (r) found=1
TID1 ⟶ Ending thread firstItem=2, lastItem=4
TID0 ⟶ Starting thread firstItem=0, lastItem=2
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ File: inputFile2.txt, (r) found=0
TID0 ⟶ Ending thread firstItem=0, lastItem=2
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./threadsafe 4 r
Main ⟶ Search Engine searching for (r) in 4 files, using 4 threads (with threshold=2)
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3
```

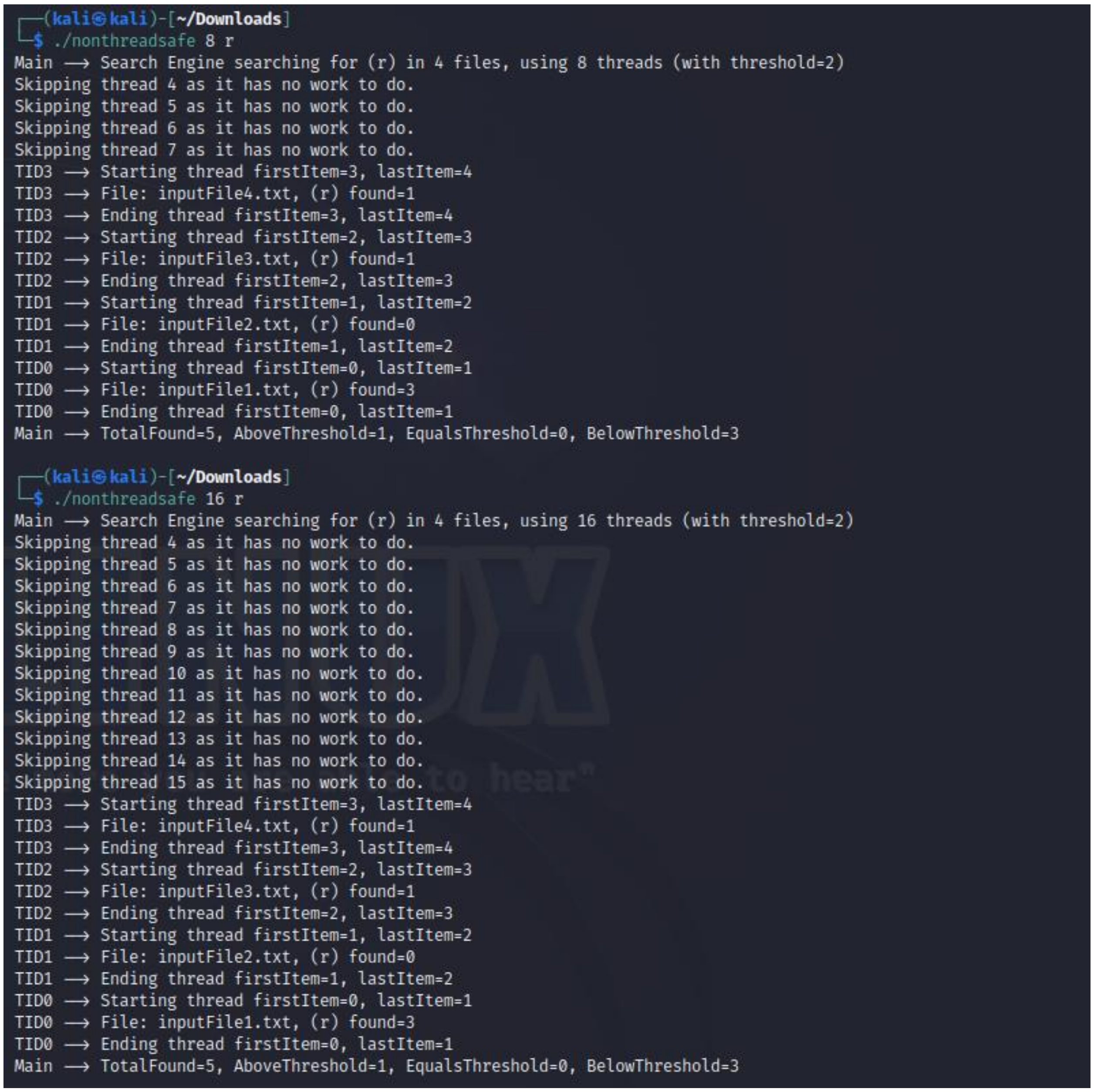


```
┌──(kali㉿kali)-[~/Downloads]
└─$ ./threadsafe 8 r
Main ⟶ Search Engine searching for (r) in 4 files, using 8 threads (with threshold=2)
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./threadsafe 16 r
Main ⟶ Search Engine searching for (r) in 4 files, using 16 threads (with threshold=2)
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
Skipping thread 8 as it has no work to do.
Skipping thread 9 as it has no work to do.
Skipping thread 10 as it has no work to do.
Skipping thread 11 as it has no work to do.
Skipping thread 12 as it has no work to do.
Skipping thread 13 as it has no work to do.
Skipping thread 14 as it has no work to do.
Skipping thread 15 as it has no work to do.
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3
```

testThreadNonSafe:

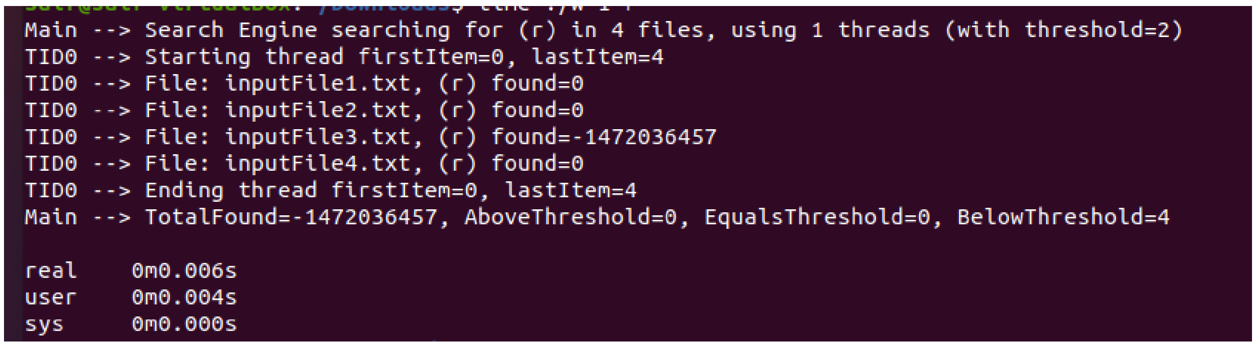


```
┌──(kali㉿kali)-[~/Downloads]
└─$ g++ nonthreadSafe.cpp -o nonthreadsafe

┌──(kali㉿kali)-[~/Downloads]
└─$ ./nonthreadsafe 1 r
Main ⟶ Search Engine searching for (r) in 4 files, using 1 threads (with threshold=2)
TID0 ⟶ Starting thread firstItem=0, lastItem=4
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ File: inputFile2.txt, (r) found=0
TID0 ⟶ File: inputFile3.txt, (r) found=1
TID0 ⟶ File: inputFile4.txt, (r) found=1
TID0 ⟶ Ending thread firstItem=0, lastItem=4
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./nonthreadsafe 2 r
Main ⟶ Search Engine searching for (r) in 4 files, using 2 threads (with threshold=2)
TID1 ⟶ Starting thread firstItem=2, lastItem=4
TID1 ⟶ File: inputFile3.txt, (r) found=1
TID1 ⟶ File: inputFile4.txt, (r) found=1
TID1 ⟶ Ending thread firstItem=2, lastItem=4
TID0 ⟶ Starting thread firstItem=0, lastItem=2
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ File: inputFile2.txt, (r) found=0
TID0 ⟶ Ending thread firstItem=0, lastItem=2
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./nonthreadsafe 4 r
Main ⟶ Search Engine searching for (r) in 4 files, using 4 threads (with threshold=2)
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3
```

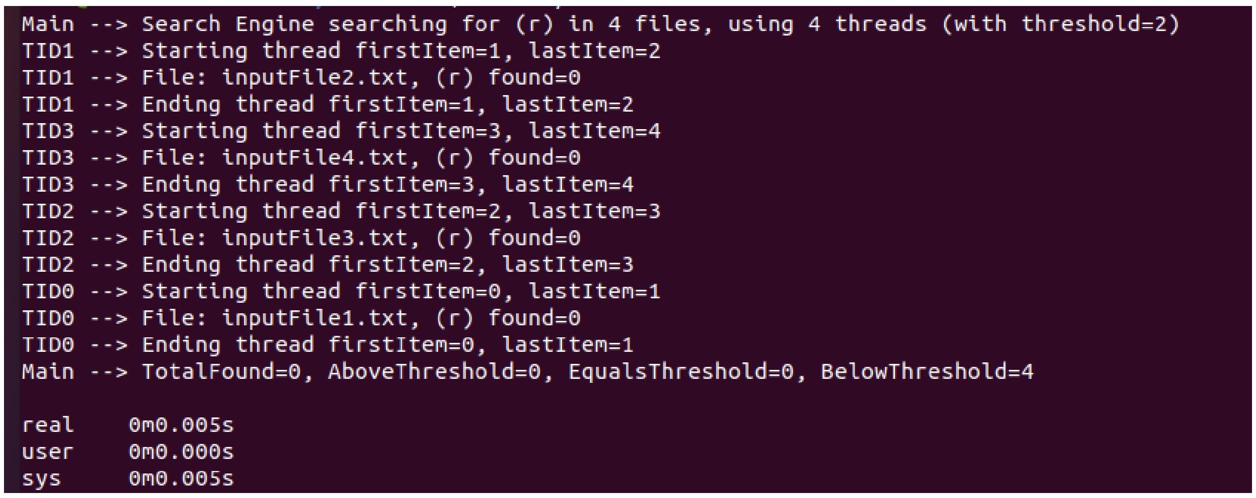


```
┌──(kali㉿kali)-[~/Downloads]
└─$ ./nonthreadsafe 8 r
Main ⟶ Search Engine searching for (r) in 4 files, using 8 threads (with threshold=2)
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3

┌──(kali㉿kali)-[~/Downloads]
└─$ ./nonthreadsafe 16 r
Main ⟶ Search Engine searching for (r) in 4 files, using 16 threads (with threshold=2)
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
Skipping thread 8 as it has no work to do.
Skipping thread 9 as it has no work to do.
Skipping thread 10 as it has no work to do.
Skipping thread 11 as it has no work to do.
Skipping thread 12 as it has no work to do.
Skipping thread 13 as it has no work to do.
Skipping thread 14 as it has no work to do.
Skipping thread 15 as it has no work to do.
TID3 ⟶ Starting thread firstItem=3, lastItem=4
TID3 ⟶ File: inputFile4.txt, (r) found=1
TID3 ⟶ Ending thread firstItem=3, lastItem=4
TID2 ⟶ Starting thread firstItem=2, lastItem=3
TID2 ⟶ File: inputFile3.txt, (r) found=1
TID2 ⟶ Ending thread firstItem=2, lastItem=3
TID1 ⟶ Starting thread firstItem=1, lastItem=2
TID1 ⟶ File: inputFile2.txt, (r) found=0
TID1 ⟶ Ending thread firstItem=1, lastItem=2
TID0 ⟶ Starting thread firstItem=0, lastItem=1
TID0 ⟶ File: inputFile1.txt, (r) found=3
TID0 ⟶ Ending thread firstItem=0, lastItem=1
Main ⟶ TotalFound=5, AboveThreshold=1, EqualsThreshold=0, BelowThreshold=3
```

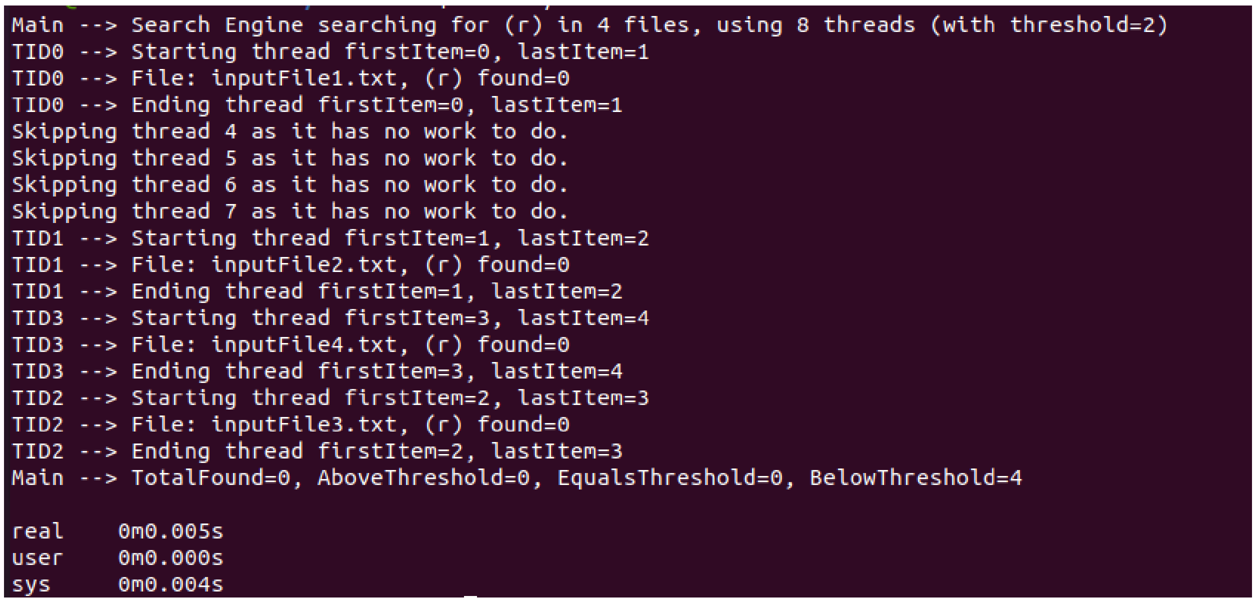
Time :



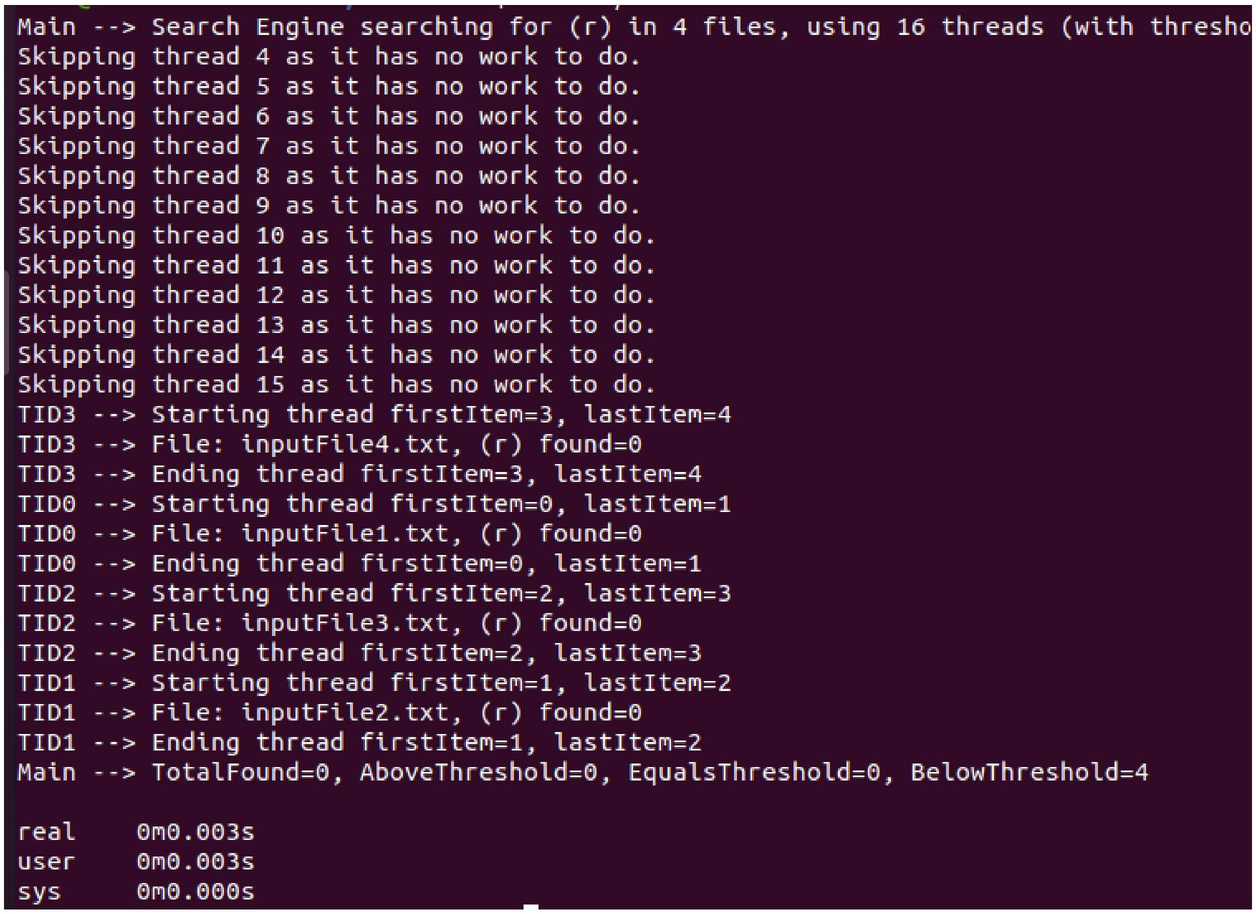
```
Main --> Search Engine searching for (r) in 4 files, using 1 threads (with threshold=2)
TID0 --> Starting thread firstItem=0, lastItem=4
TID0 --> File: inputFile1.txt, (r) found=0
TID0 --> File: inputFile2.txt, (r) found=0
TID0 --> File: inputFile3.txt, (r) found=-1472036457
TID0 --> File: inputFile4.txt, (r) found=0
TID0 --> Ending thread firstItem=0, lastItem=4
Main --> TotalFound=-1472036457, AboveThreshold=0, EqualsThreshold=0, BelowThreshold=4

real    0m0.006s
user    0m0.004s
sys     0m0.000s
```

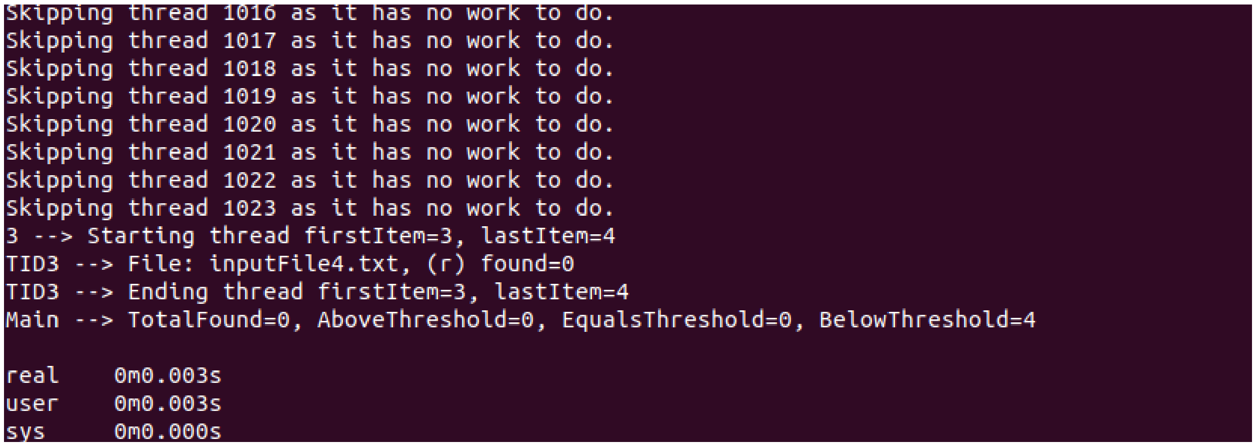


```
Main --> Search Engine searching for (r) in 4 files, using 4 threads (with threshold=2)
TID1 --> Starting thread firstItem=1, lastItem=2
TID1 --> File: inputFile2.txt, (r) found=0
TID1 --> Ending thread firstItem=1, lastItem=2
TID3 --> Starting thread firstItem=3, lastItem=4
TID3 --> File: inputFile4.txt, (r) found=0
TID3 --> Ending thread firstItem=3, lastItem=4
TID2 --> Starting thread firstItem=2, lastItem=3
TID2 --> File: inputFile3.txt, (r) found=0
TID2 --> Ending thread firstItem=2, lastItem=3
TID0 --> Starting thread firstItem=0, lastItem=1
TID0 --> File: inputFile1.txt, (r) found=0
TID0 --> Ending thread firstItem=0, lastItem=1
Main --> TotalFound=0, AboveThreshold=0, EqualsThreshold=0, BelowThreshold=4

real    0m0.005s
user    0m0.000s
sys     0m0.005s
```



```
Main --> Search Engine searching for (r) in 4 files, using 8 threads (with threshold=2)
TID0 --> Starting thread firstItem=0, lastItem=1
TID0 --> File: inputFile1.txt, (r) found=0
TID0 --> Ending thread firstItem=0, lastItem=1
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
TID1 --> Starting thread firstItem=1, lastItem=2
TID1 --> File: inputFile2.txt, (r) found=0
TID1 --> Ending thread firstItem=1, lastItem=2
TID3 --> Starting thread firstItem=3, lastItem=4
TID3 --> File: inputFile4.txt, (r) found=0
TID3 --> Ending thread firstItem=3, lastItem=4
TID2 --> Starting thread firstItem=2, lastItem=3
TID2 --> File: inputFile3.txt, (r) found=0
TID2 --> Ending thread firstItem=2, lastItem=3
Main --> TotalFound=0, AboveThreshold=0, EqualsThreshold=0, BelowThreshold=4

real    0m0.005s
user    0m0.000s
sys     0m0.004s
```


```
Main --> Search Engine searching for (r) in 4 files, using 16 threads (with thresho
Skipping thread 4 as it has no work to do.
Skipping thread 5 as it has no work to do.
Skipping thread 6 as it has no work to do.
Skipping thread 7 as it has no work to do.
Skipping thread 8 as it has no work to do.
Skipping thread 9 as it has no work to do.
Skipping thread 10 as it has no work to do.
Skipping thread 11 as it has no work to do.
Skipping thread 12 as it has no work to do.
Skipping thread 13 as it has no work to do.
Skipping thread 14 as it has no work to do.
Skipping thread 15 as it has no work to do.
TID3 --> Starting thread firstItem=3, lastItem=4
TID3 --> File: inputFile4.txt, (r) found=0
TID3 --> Ending thread firstItem=3, lastItem=4
TID0 --> Starting thread firstItem=0, lastItem=1
TID0 --> File: inputFile1.txt, (r) found=0
TID0 --> Ending thread firstItem=0, lastItem=1
TID2 --> Starting thread firstItem=2, lastItem=3
TID2 --> File: inputFile3.txt, (r) found=0
TID2 --> Ending thread firstItem=2, lastItem=3
TID1 --> Starting thread firstItem=1, lastItem=2
TID1 --> File: inputFile2.txt, (r) found=0
TID1 --> Ending thread firstItem=1, lastItem=2
Main --> TotalFound=0, AboveThreshold=0, EqualsThreshold=0, BelowThreshold=4

real    0m0.003s
user    0m0.003s
sys     0m0.000s
```


```
Skipping thread 1016 as it has no work to do.
Skipping thread 1017 as it has no work to do.
Skipping thread 1018 as it has no work to do.
Skipping thread 1019 as it has no work to do.
Skipping thread 1020 as it has no work to do.
Skipping thread 1021 as it has no work to do.
Skipping thread 1022 as it has no work to do.
Skipping thread 1023 as it has no work to do.
3 --> Starting thread firstItem=3, lastItem=4
TID3 --> File: inputFile4.txt, (r) found=0
TID3 --> Ending thread firstItem=3, lastItem=4
Main --> TotalFound=0, AboveThreshold=0, EqualsThreshold=0, BelowThreshold=4

real    0m0.003s
user    0m0.003s
sys     0m0.000s
```