Eric Gonzalez

CSC 415

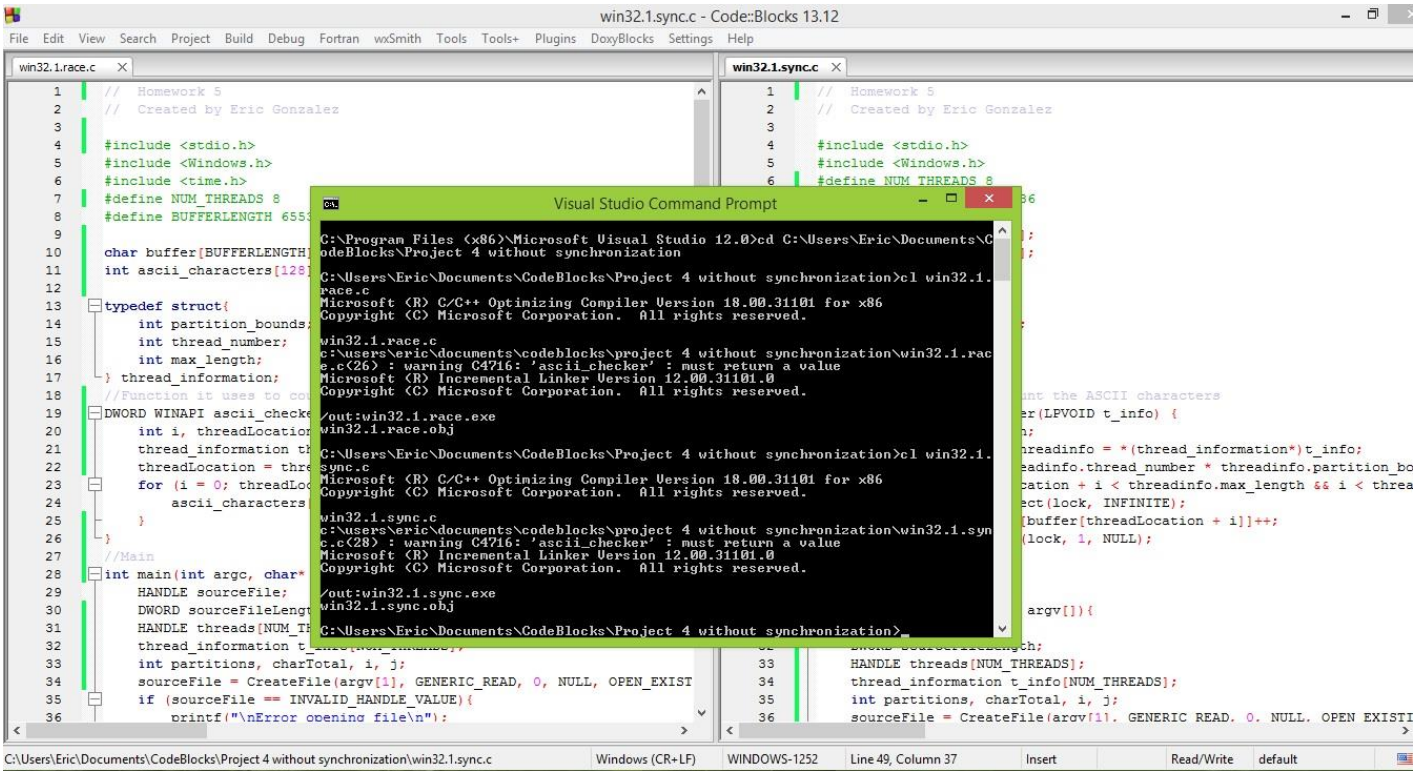
Homework #5

We had 2 parts for this assignment. The first part had us write two versions of win32 and posix code. One version required us to update our existing posix and win32 code from homework 4 to use a one-dimensional global array to hold all of the character counts. This allows us to increment the same array without using synchronization. The second version required that we protect the increment of the global array using a mutex.

Part 1:

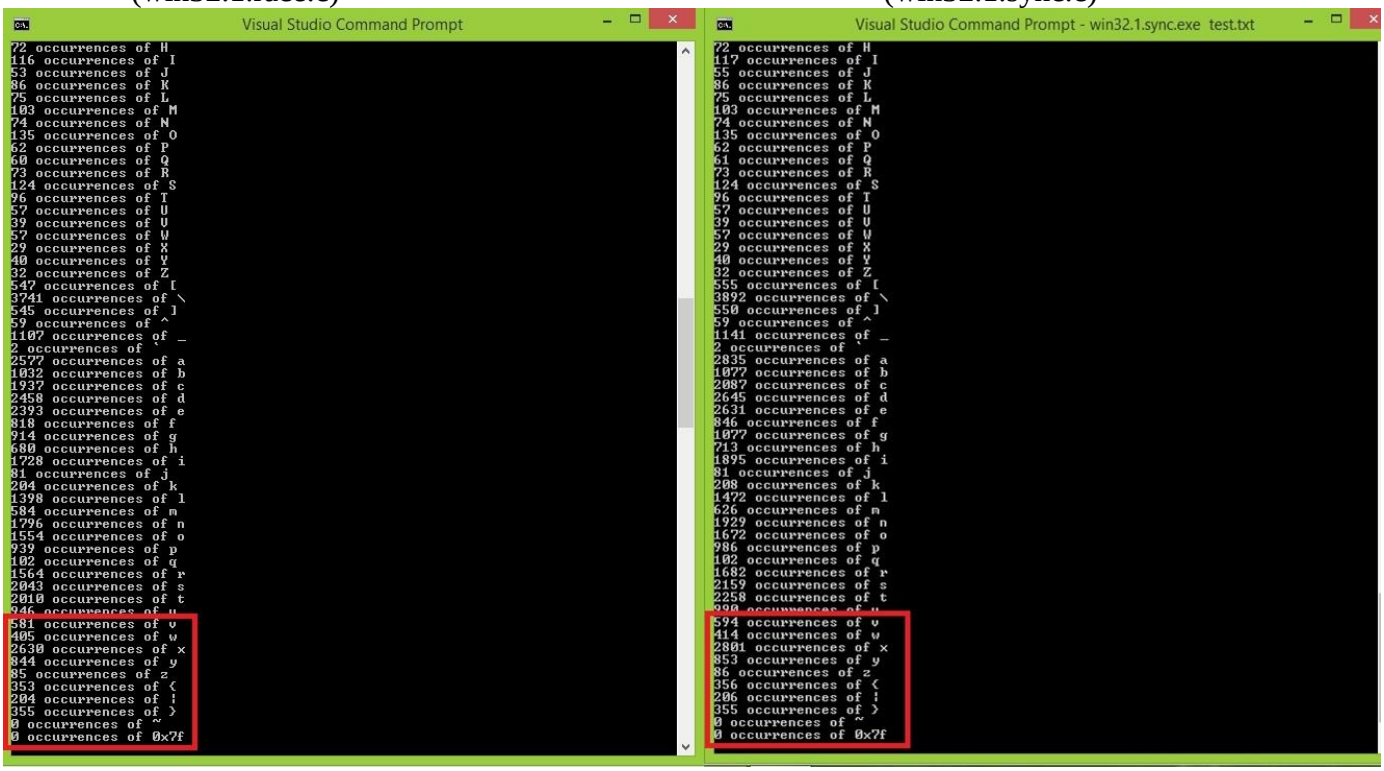
Win32

-First I compiled both programs…



-Next I executed the two programs using the same test.txt file. The test file has random input as well as website source code to see a difference in counts…

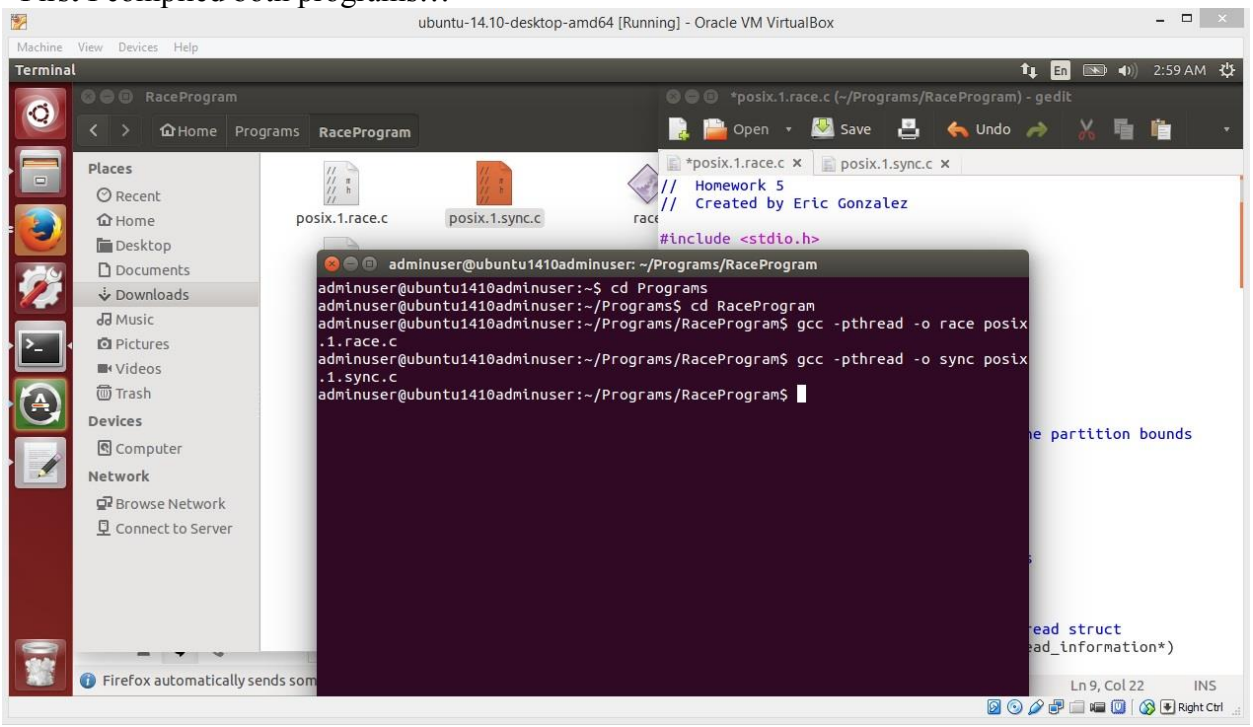
(win32.1.race.c) (win32.1.sync.c)



-When comparing the counts you can see that *win32.1.sync.c* has larger numbers than *win32.1.race.c* , its clearly seen in the sample results I boxed. This is the case because in *win32.1.sync.c* we use a mutex, meaning we ensure that no two concurrent processes are in their critical condition at the same time therefore preventing race conditions, something we see in *win32.1.race.c* because the output is dependent on the sequence or timing of other uncontrollable events, in our case essentially counting the wrong number of occurrences of a certain ASCII character.

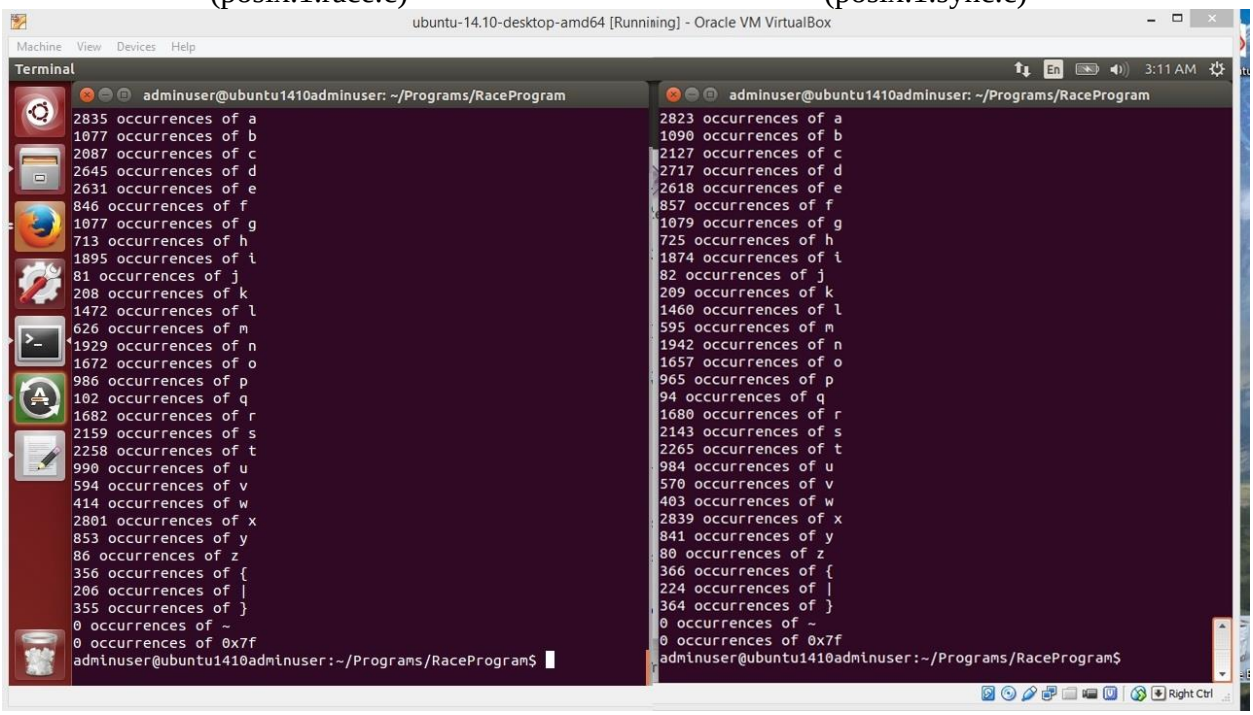
Posix

- First I compiled both programs…



-Next I executed them using the same test.txt file I used for win32…

(posix.1.race.c) (posix.1.sync.c)

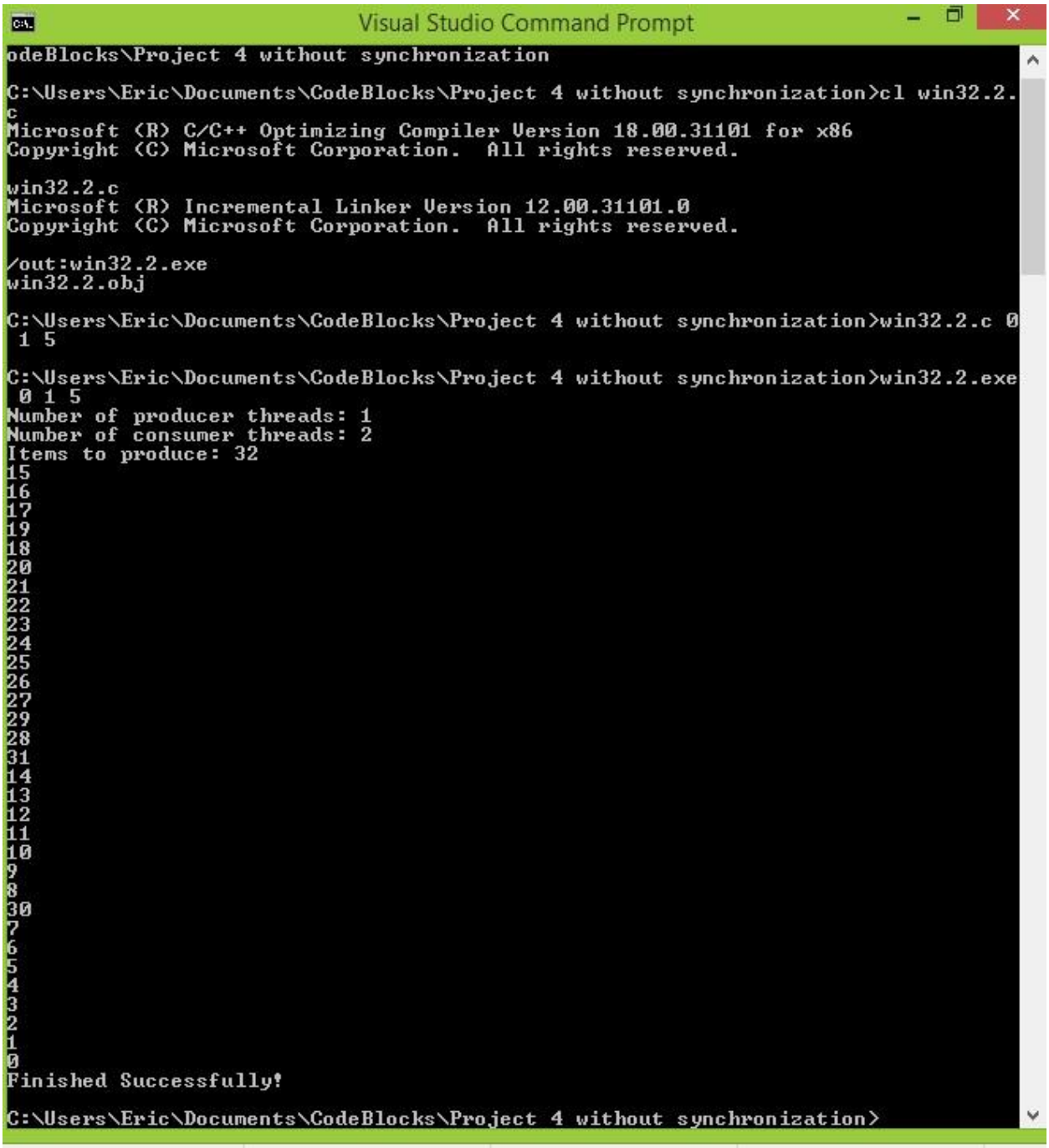


-When comparing the counts you can see that *posix.1.sync.c* has larger numbers than *posix.1.race.c*. Should be the same explanation I gave above for the win32 case.

Part 2:

Win32

-First I compiled the program then executed with 1 producer, 2 consumer, and 32 items to produce…



-then executed the program with 2 producers, 1 consumer and 32 items to produce…

```
C:\Users\Eric\Documents\CodeBlocks\Project 4 without synchronization>win32.2.exe
 1 0 5
Number of producer threads: 2
Number of consumer threads: 1
Items to produce: 32
17
1
31
2
3
4
5
6
7
8
9
10
11
12
13
14
15
30
29
28
27
26
25
24
23
22
21
20
19
18
16
0
Finished Successfully!
```

-then again executed the program with 2 producers, 4 consumers and 32 items to produce…

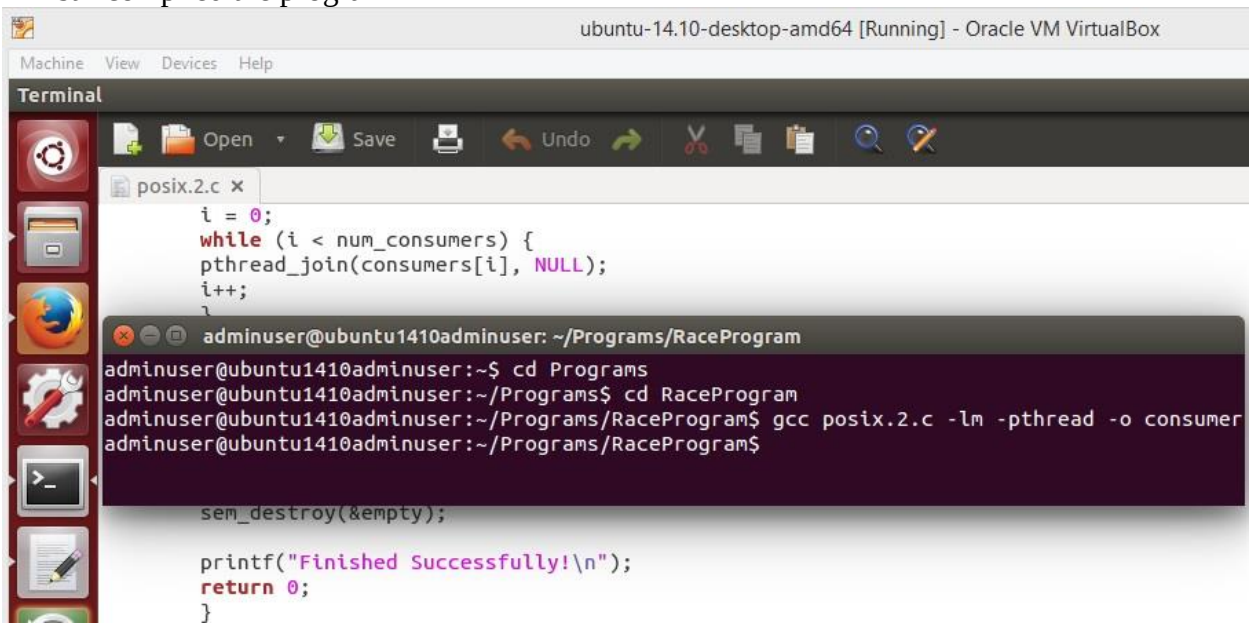
```
C:\Users\Eric\Documents\CodeBlocks\Project 4 without synchronization>win32.2.exe
 1 2 5
Number of producer threads: 2
Number of consumer threads: 4
Items to produce: 32
19
31
1
5
3
7
2
9
10
11
12
13
14
15
4
30
29
28
27
26
25
8
24
23
22
21
20
6
18
17
16
0
Finished Successfully!
```

-then finally executed the program with 4 consumers, 2 producers and 32 items to produce…

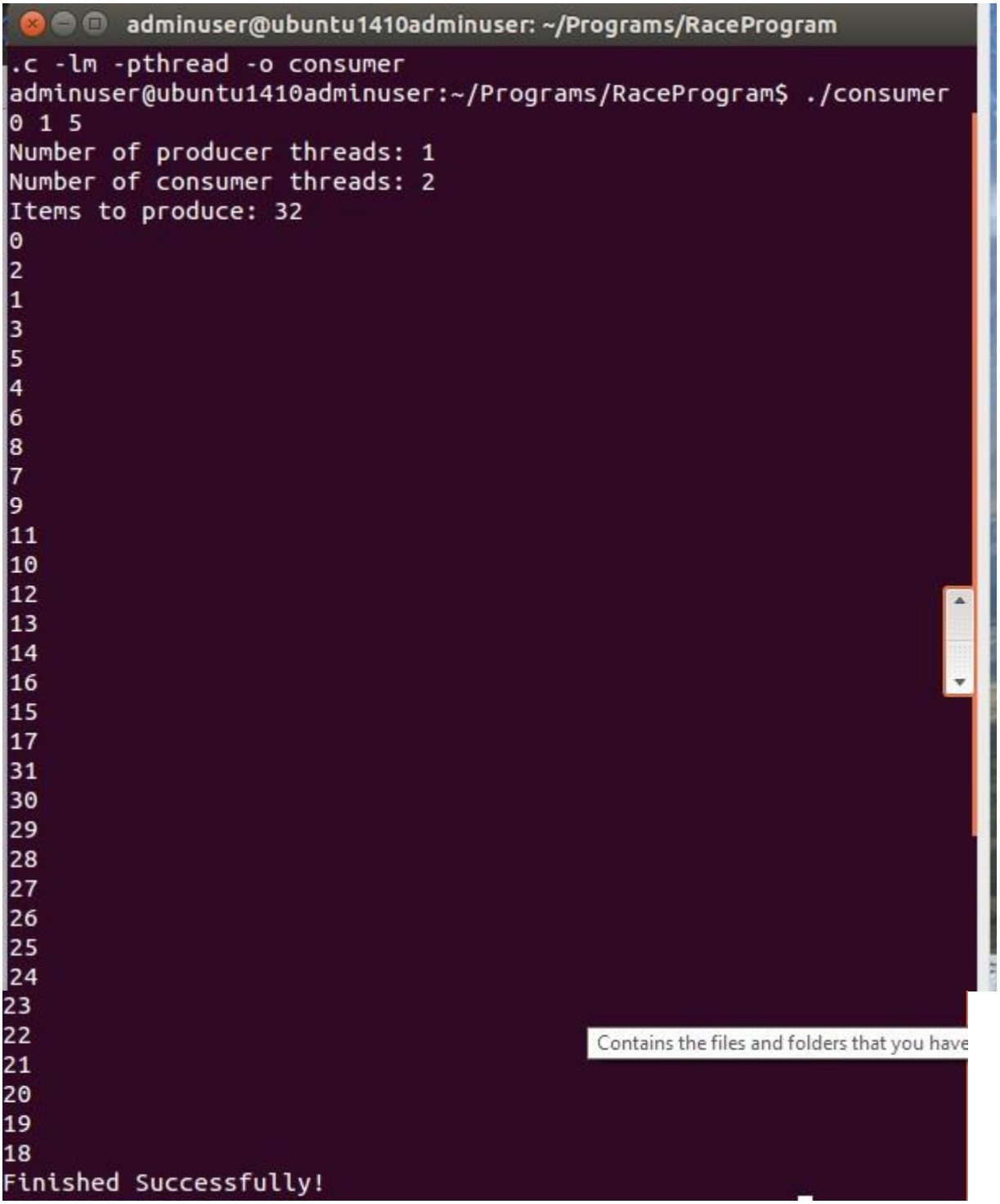
```
C:\Users\Eric\Documents\CodeBlocks\Project 4 without synchronization>win32.2.exe
 2 1 5
Number of producer threads: 4
Number of consumer threads: 2
Items to produce: 32
15
16
24
25
18
26
19
27
17
20
21
29
22
30
23
28
14
13
12
11
10
9
31
7
8
6
5
4
3
2
1
0
Finished Successfully!
```

Posix

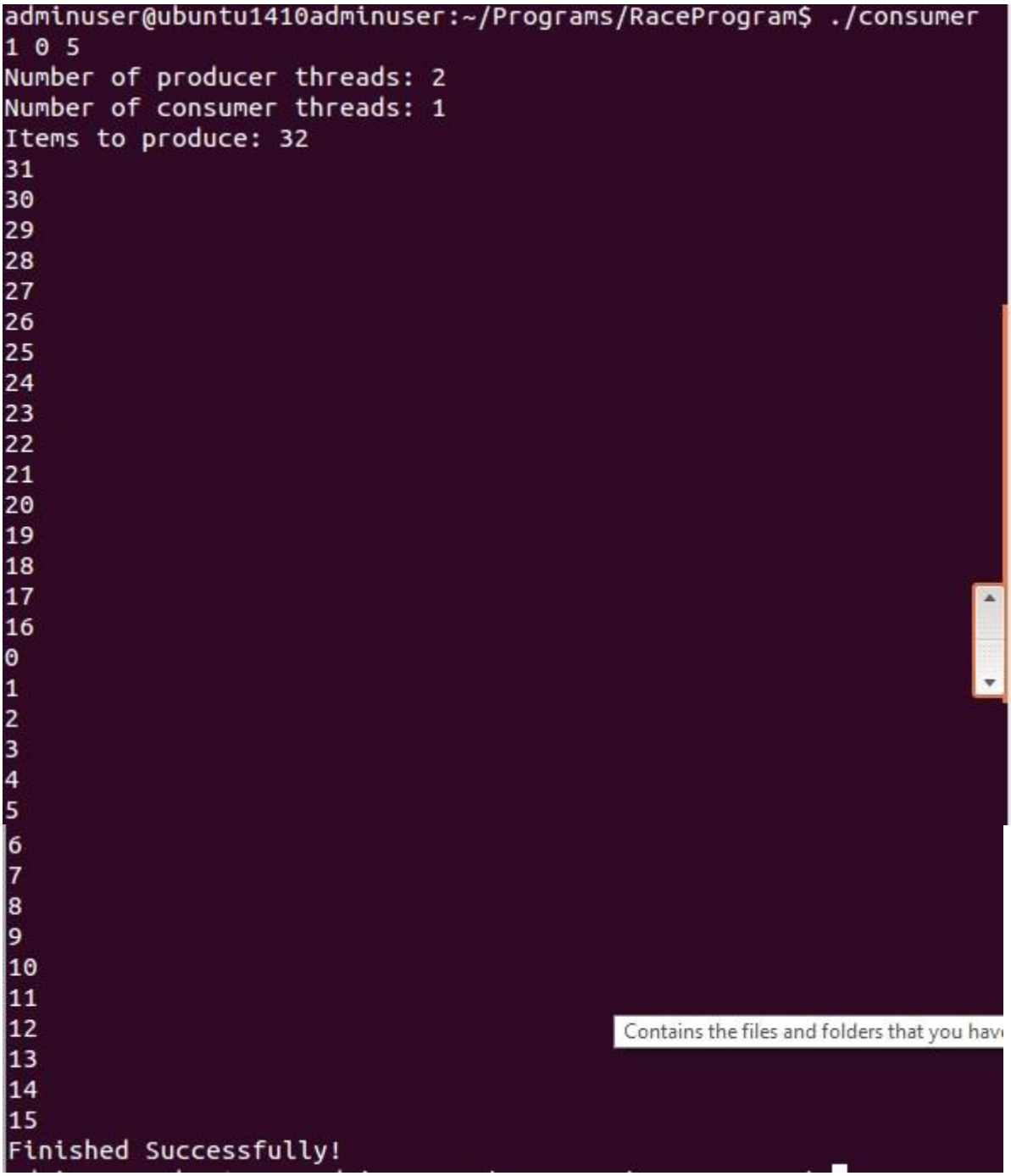
- First I compiled the program



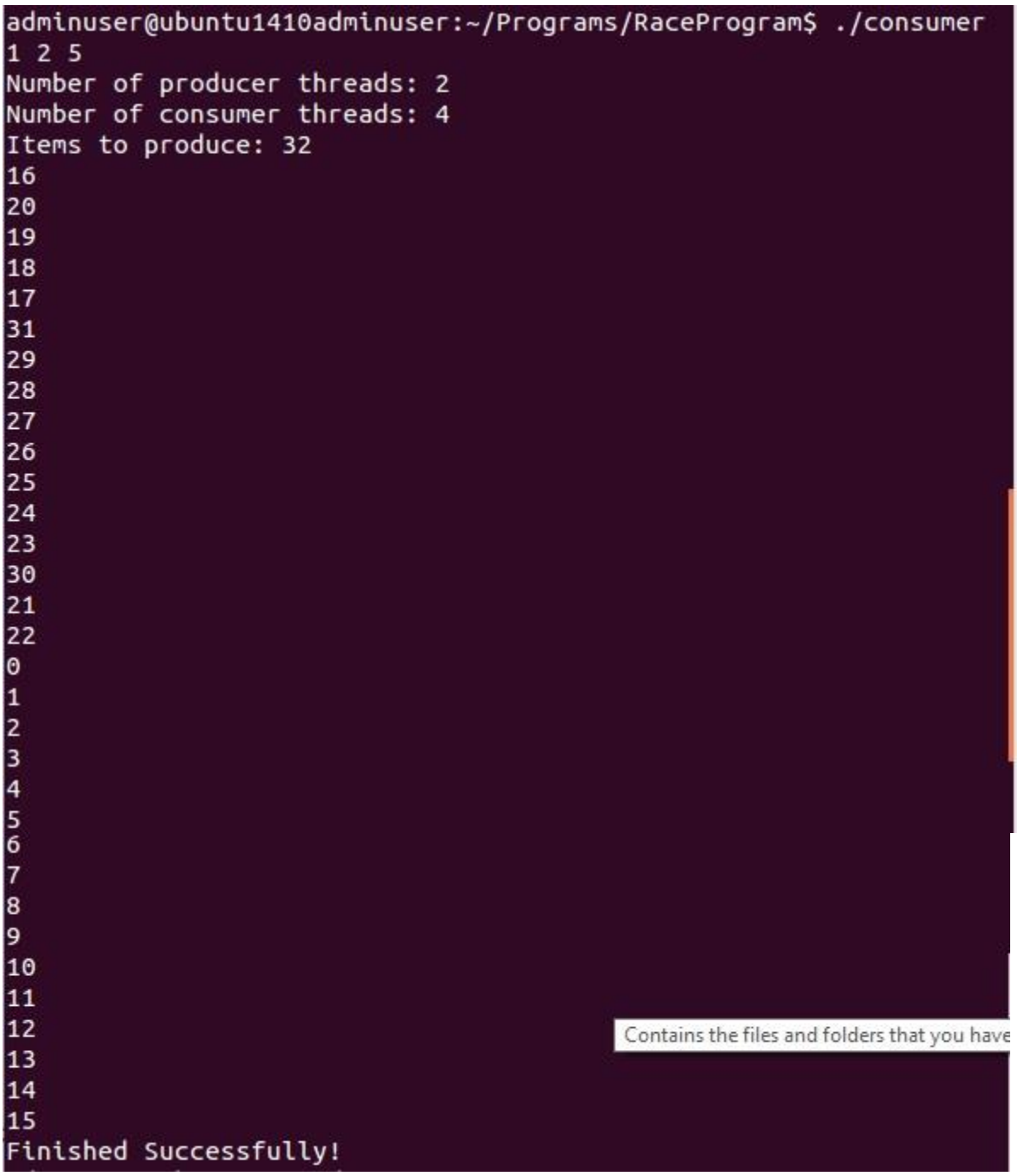
-then I executed the program with 1 producer, 2 consumer, and 32 items to produce…



-then executed the program with 2 producers, 1 consumer and 32 items to produce…



-then again executed the program with 2 producers, 4 consumers and 32 items to produce…



-then finally executed the program with 4 consumers, 2 producers and 32 items to produce…

```
adminuser@ubuntu1410adminuser:~/Programs/RaceProgram$ ./consumer
2 1 5
Number of producer threads: 4
Number of consumer threads: 2
Items to produce: 32
31
30
29
28
27
25
24
26
23
22
21
20
19
17
16
18
15
13
14
12
11
10
10
9
8
0
1
2
3
4
5
6
7
Finished Successfully!
```