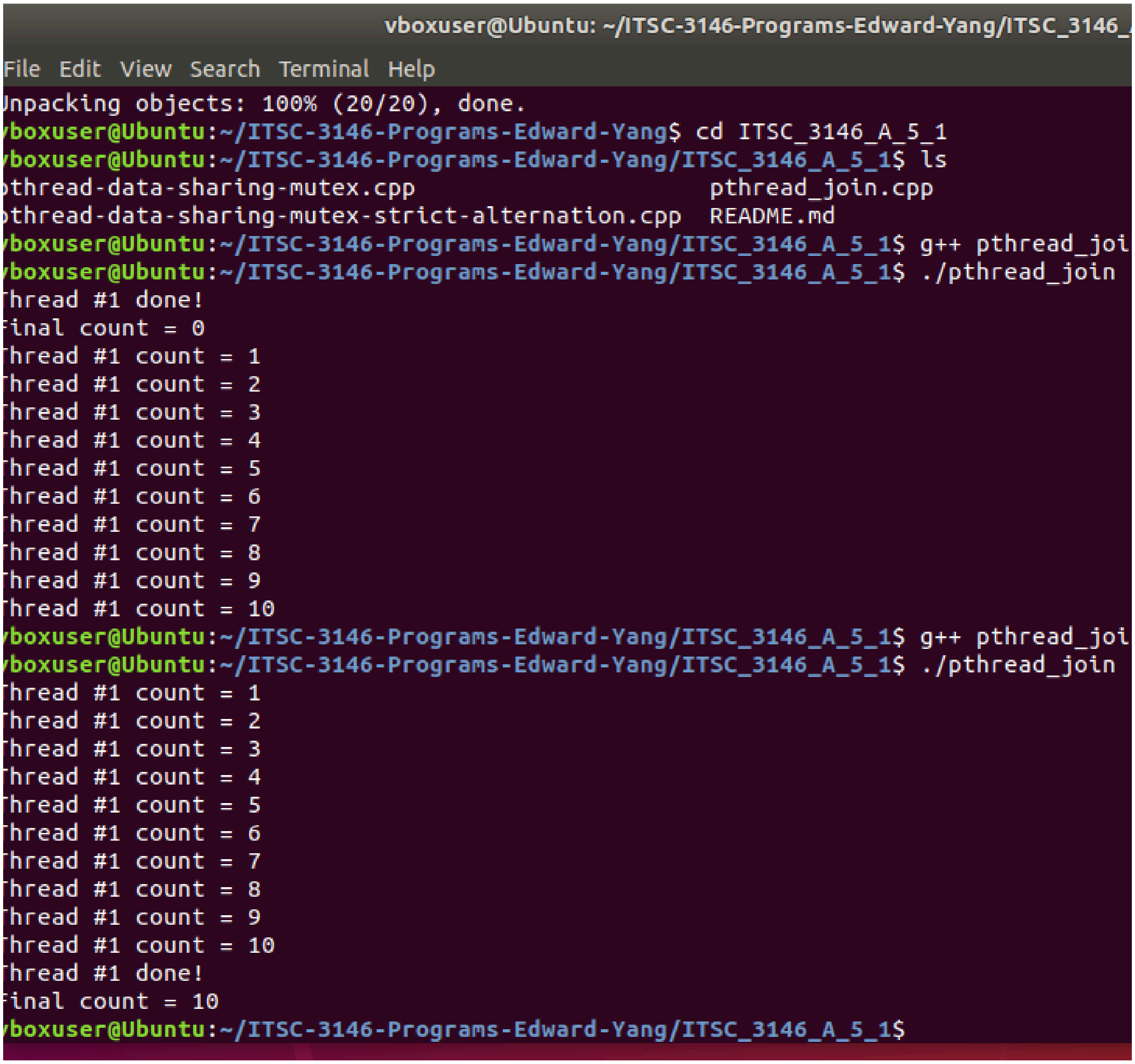
Part 1



vboxuser@Ubuntu: ~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_
File Edit View Search Terminal Help

```
Unpacking objects: 100% (20/20), done.
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang$ cd ITSC_3146_A_5_1
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ ls
pthread-data-sharing-mutex.cpp                    pthread_join.cpp
pthread-data-sharing-mutex-strict-alternation.cpp  README.md
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ g++ pthread_joi
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ ./pthread_join
Thread #1 done!
Final count = 0
Thread #1 count = 1
Thread #1 count = 2
Thread #1 count = 3
Thread #1 count = 4
Thread #1 count = 5
Thread #1 count = 6
Thread #1 count = 7
Thread #1 count = 8
Thread #1 count = 9
Thread #1 count = 10
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ g++ pthread_joi
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ ./pthread_join
Thread #1 count = 1
Thread #1 count = 2
Thread #1 count = 3
Thread #1 count = 4
Thread #1 count = 5
Thread #1 count = 6
Thread #1 count = 7
Thread #1 count = 8
Thread #1 count = 9
Thread #1 count = 10
Thread #1 done!
Final count = 10
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$
```

Part 2

i. What does it do?
   1. Creates two threads where each thread counts ten times then joins the threads together to count to 20

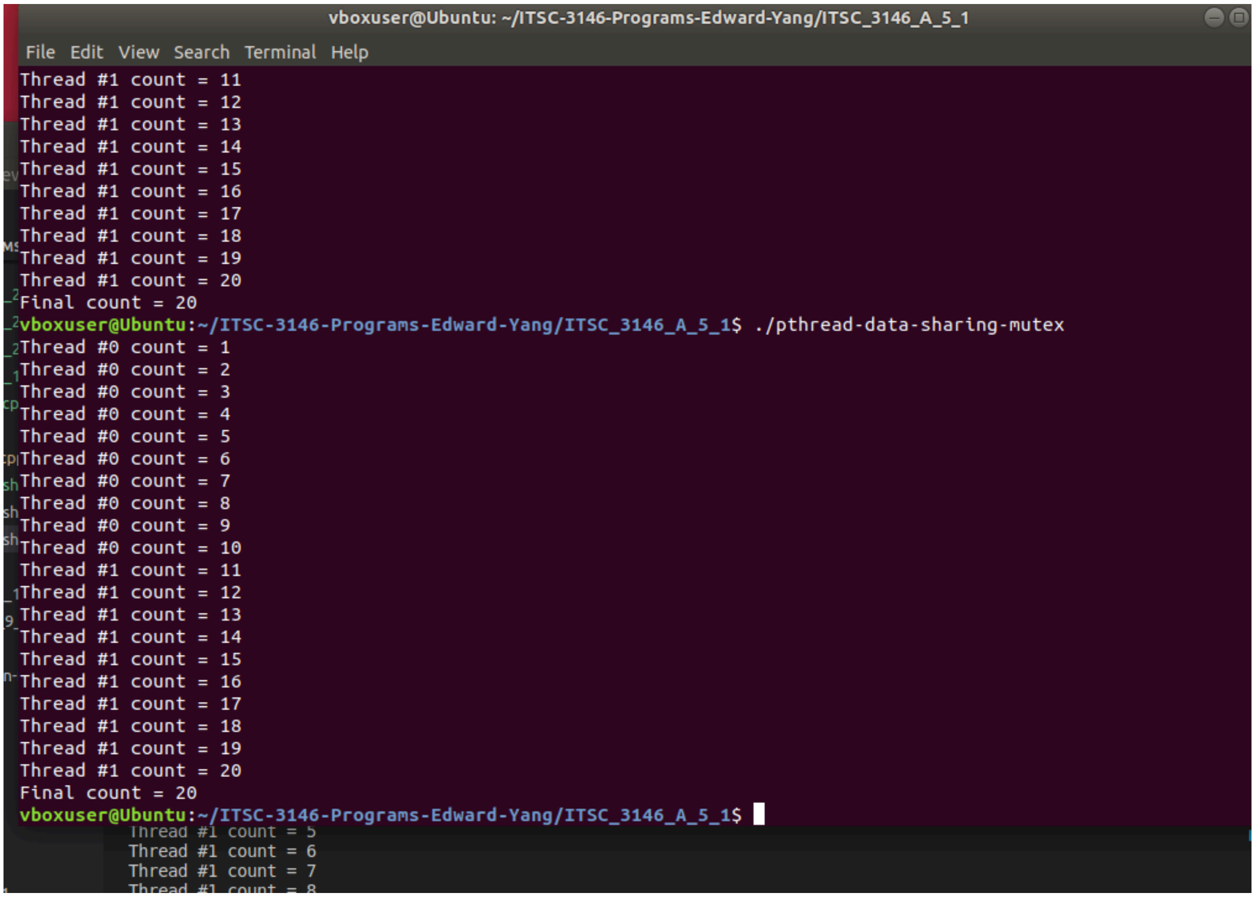
ii. What output does it produce?
   1. Should always count to 20 but will sometimes have 11 in the first thread or cound 11 twice

iii. Examine the program code carefully. Is the program functioning correctly?

        1. No lacks synchronization around the count variable and everything is happening at the same time

    iv. If you do not think that the program is working correctly, describe why?

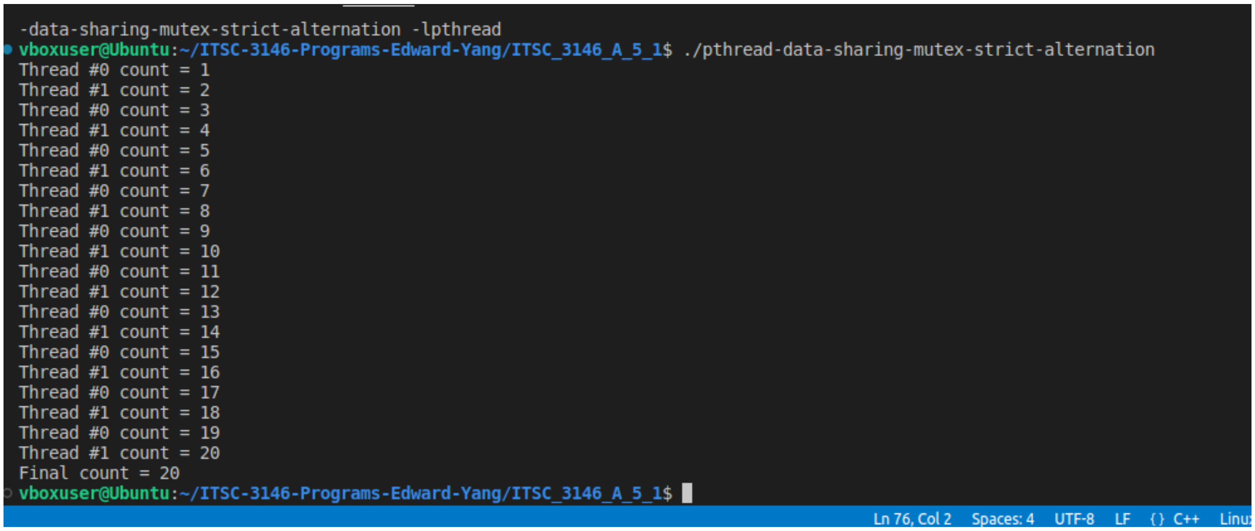
        1. The program is not functioning correctly. Each thread safely increments the shared count variable ten times. To correct this issue and ensure that the final count value is always 20, synchronization techniques such as mutexes should be employed to control access to the count variable, preventing multiple threads from modifying it simultaneously.


vboxuser@Ubuntu: ~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1
File Edit View Search Terminal Help

```
Thread #1 count = 11
Thread #1 count = 12
Thread #1 count = 13
Thread #1 count = 14
Thread #1 count = 15
Thread #1 count = 16
Thread #1 count = 17
Thread #1 count = 18
Thread #1 count = 19
Thread #1 count = 20
Final count = 20
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ ./pthread-data-sharing-mutex
Thread #0 count = 1
Thread #0 count = 2
Thread #0 count = 3
Thread #0 count = 4
Thread #0 count = 5
Thread #0 count = 6
Thread #0 count = 7
Thread #0 count = 8
Thread #0 count = 9
Thread #0 count = 10
Thread #1 count = 11
Thread #1 count = 12
Thread #1 count = 13
Thread #1 count = 14
Thread #1 count = 15
Thread #1 count = 16
Thread #1 count = 17
Thread #1 count = 18
Thread #1 count = 19
Thread #1 count = 20
Final count = 20
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ 
```

```
          Thread #1 count = 5
          Thread #1 count = 6
          Thread #1 count = 7
          Thread #1 count = 8
```

Part 3



```
-data-sharing-mutex-strict-alternation -lpthread
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$ ./pthread-data-sharing-mutex-strict-alternation
Thread #0 count = 1
Thread #1 count = 2
Thread #0 count = 3
Thread #1 count = 4
Thread #0 count = 5
Thread #1 count = 6
Thread #0 count = 7
Thread #1 count = 8
Thread #0 count = 9
Thread #1 count = 10
Thread #0 count = 11
Thread #1 count = 12
Thread #0 count = 13
Thread #1 count = 14
Thread #0 count = 15
Thread #1 count = 16
Thread #0 count = 17
Thread #1 count = 18
Thread #0 count = 19
Thread #1 count = 20
Final count = 20
vboxuser@Ubuntu:~/ITSC-3146-Programs-Edward-Yang/ITSC_3146_A_5_1$
```