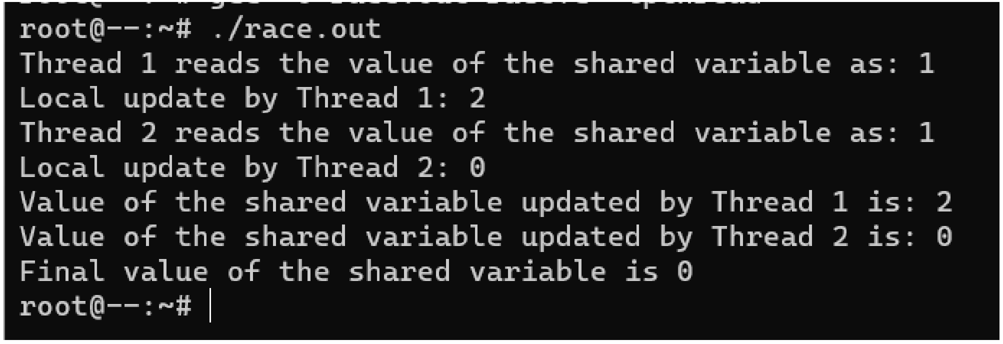
Ramzy Izza Wardhana
21/472698/PA/20322
IUP CS B – Computer Networking

Activity 9.1 – Process Synchronization and Deadlock

Compile and run the source code listing above and compare your result with your friends. Screenshot your result and answer the following questions.

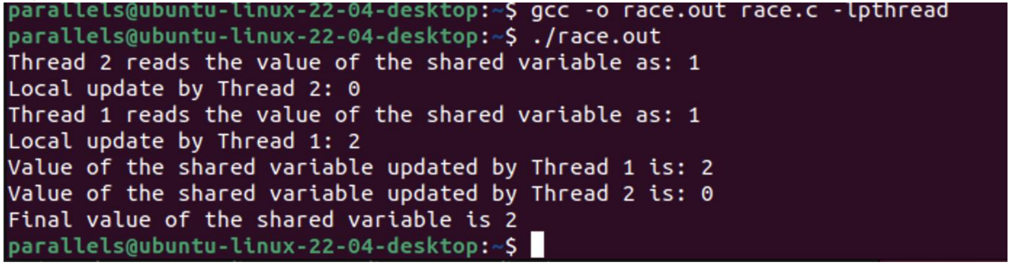
a. Is the final result of the shared variable similar to your friends?

My Result:



```
root@--:~# ./race.out
Thread 1 reads the value of the shared variable as: 1
Local update by Thread 1: 2
Thread 2 reads the value of the shared variable as: 1
Local update by Thread 2: 0
Value of the shared variable updated by Thread 1 is: 2
Value of the shared variable updated by Thread 2 is: 0
Final value of the shared variable is 0
root@--:~#
```

My Friends Result:



```
parallels@ubuntu-linux-22-04-desktop:~$ gcc -o race.out race.c -lpthread
parallels@ubuntu-linux-22-04-desktop:~$ ./race.out
Thread 2 reads the value of the shared variable as: 1
Local update by Thread 2: 0
Thread 1 reads the value of the shared variable as: 1
Local update by Thread 1: 2
Value of the shared variable updated by Thread 1 is: 2
Value of the shared variable updated by Thread 2 is: 0
Final value of the shared variable is 2
parallels@ubuntu-linux-22-04-desktop:~$
```

The final result (0) is differ with my friend's result (2)

b. Why does this happen?

This happens since different devices may have different execution order. In my case, the function increment() runs first by reading the value 1 and increment it by 1 (1 + 1 = 2) which then followed by decrement() function to also read 1 as the initial value and decrement it by 1 (1 – 1 = 0). As a result, the final value that I got is 0.

On the other hand, my friends final result is 2. This was caused by execution order of decrement() function runs first by reading the initial value as 1 (1 – 1 = 0). Then increment() function also read the initial value as 1 to and increment it by 1 (1+1 = 2). Therefore, the final result of my friend's code is 2.

c. Which part of the source code list above is the critical section?

The critical section in the function above is located when the program needs to update the value, such as x++ and y--.