

Part 2: Pthread Mutex Observation (5 points)

1. A file named **pthread-data-sharing-mutex-os-call.cpp** has been provided to you in the same project.
2. Compile the program and **execute it several times**, at least 10. Make sure to **pay close attention to the output that the program produces**.
 - a. Create a Word or Google Docs document.
 - b. In this document, answer the following questions about the program's behavior:
 - i. What does it do?

This program creates 4 separate threads that are supposed to incrementally count to 40.

- ii. What output does it produce?

```
vboxuser@Ubuntu: /media/sf_ITSC-3146/ITSC_3146_A_5_2$ ./oscall
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 = 12
Thread #3 count = 12
Thread #3 count = 13
Thread #3 count = 14
Thread #3 count = 15
Thread #3 count = 16
Thread #3 count = 17
Thread #3 count = 18
Thread #3 count = 19
Thread #3 count = 20
Thread #3 count = 21
Thread #2 count = 21
Thread #2 count = 22
Thread #2 count = 23
Thread #2 count = 24
Thread #2 count = 25
Thread #2 count = 26
Thread #2 count = 27
Thread #2 count = 28
Thread #2 count = 29
Thread #2 count = 30
Thread #0 count = 31
Thread #0 count = 32
Thread #0 count = 33
Thread #0 count = 34
Thread #0 count = 35
Thread #0 count = 36
Thread #0 count = 37
Thread #0 count = 38
Thread #0 count = 39
Thread #0 count = 40
Final count = 40
```

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

The Program is not quite functioning properly.

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

The problem right now is that the mutex functionality is not being implemented correctly. There is no locking and unlocking in the critical sections, which is why the threads are executing 1 after another in the output.