

Assignment 11

Programming Languages

1. [60 pts, concurrency] Consider the following Python program:

```
from threading import Thread, Lock, current_thread
from time import sleep
status, status_lock, count, count_lock = 1, Lock(), 1, Lock()

def fun1():
    global count, status
    while 1:
        with count_lock:
            print (current_thread().name, 'count lock acquired') # line 10
            count += 1
        with status_lock:
            print (current_thread().name, 'status lock acquired') # line 13
            status = count
            print (current_thread().name, 'updated status= %d' % (status))
        sleep(0.5)

def fun2():
    global count, status
    while 1:
        with status_lock:
            print (current_thread().name, 'status lock acquired') # line 22
            count += 1
        with count_lock:
            print (current_thread().name, 'count lock acquired') # line 25
            status = count
            print (current_thread().name, 'updated status= %d' % (status))
        sleep(0.5)

if __name__ == "__main__":
    f1 = Thread(target=fun1)
    f2 = Thread(target=fun2)
    f1.start()
    f2.start()
```

- (a.) Run it and observe that the program hangs shortly after it started running. Explain what is happening and what causes the hang?
- (b.) Comment lines 10, 13, 22 and 25 and run it again. Explain what is happening and why it stopped hanging? Or did it?
- (c.) Explain how would you fix the problem?
2. [40 pts, concurrency] Imagine you need a producer/consumer design with multiple producers and consumers to process some data. Sketch a design (pseudocode) that uses a counting semaphore for critical section management.

