**Victor Chu**

**October 17, 2016**

**HW 4**

1. a: (0, 0, 0)

b: (1, 0, 1)

c: (1, 1, 2)

d: (1, 2, 4)

e: (2, 2, 4)

f: (0, 2, 0)

2.

(a) a, b, c, d: x = 8, y = 4

a, c, b, d: x = 4, y = 2

a, c, d, b: x = 8, y = 9

c, d, a, b: x = 1, y = 2

c, a, d, b: x = 8, y = 9

c, a, b, d: x = 4, y = 2

(b) a, b, c, d

c, d, a, b

These executions still work because they run all of one thread before running the other.

3. Let’s say that a reader starts off by getting a lock. Let’s also assume that a writer request the same lock. The writer is now waiting for the reader to finish before it can obtain the lock and continue its process. However, if we also assume that a reader will be continually added to some queue every time the current reader is halfway into its process, then the writer process has reached starvation because the read/write lock will keep giving access to the next reader in the queue since the current reader is still holding the lock. The writer will never get the lock since it has to wait for all readers to release the lock before obtaining it. The writer process has reached starvation.

4. We have to consider three factors when evaluating the performance of fine-grained vs. coarse-grained locking: lock overhead (time to create/get/release locks), lock contention (wait time required when a process tries to get a lock that another process already has), and deadlock (when a process is waiting on a lock forever because another process never releases it).

If we have coarse-grained locking, then the single-process performance is better because it will requires less locks, which means less lock overhead. However, the multi-process performance will be worse because there is more lock contention, since there are fewer locks to chose from and therefore results in more processes using the same lock. Also, in this case where there is more lock overlap, there is an increased chance of deadlock because lock overlap increases the chance of circular dependencies; this also decreases performance for multi-process systems.

If we have fine-grained locking, then the single-process performance is worse since we have more locks and therefore more lock overhead. However, the multi-process performance is better because there is less lock contention, since there are more locks to choose from, and therefore requires less lock overlap. Moreover, there is less chance of deadlock in this case since there is less lock overlap.