Midterm Practice Problems
CSc 422, Fall 2020

September 30, 2020

Concurrency

Consider the following code fragment. Give all possible outputs and clearly and concisely explain your answer (continue your answer on the next page if needed). Note that in this example threads call functions with zero parameters.

```
int x;
Lock l;
Semaphore s = 0;

void foo( ) {
    x = x + 1;
    V(s);
}

void bar( ) {
    P(s);
    P(s);
    x = x * 2;
}

void baz( ) {
    Acquire(l);
    x = x + 2;
    V(s);
    Release(l);
}

main() {
    x = 0;
    Thread t, u, v;
    ThreadCreate(&t, foo);      // fork t; t calls foo when it runs
    ThreadCreate(&u, bar);
    ThreadCreate(&v, baz);
    ThreadJoin(&t); ThreadJoin(&u); ThreadJoin(&v);      // block until threads complete
    print(x);
}
```

Semaphores

The following is a correct two-thread symmetric barrier using semaphores.

```
// s is a two element array, with indices 0 and 1
sem s[2] = {0,0}
Thread 0: V(s[0]); P(s[1])
Thread 1: V(s[1]); P(s[0])
```

A. Suppose we instead propose the following solution:

```
// s is a two element array, with indices 0 and 1
sem s[2] = {0,0}
Thread 0: P(s[1]); V(s[0])
Thread 1: P(s[0]); V(s[1])
```

Is this solution correct? Briefly explain.

B. Suppose we instead propose the following solution:

```
// s is a two element array, with indices 0 and 1
sem s[2] = {0,0}
Thread 0: V(s[0]); P(s[1])
Thread 1: P(s[0]); V(s[1])
```

Is this solution correct? Briefly explain.

Consider a proposed four-thread symmetric barrier.

```
// s is a four element array, with indices 0, 1, 2, and 3
sem s[4] = {0,0,0,0}
Thread 0: V(s[0]); P(s[1]); V(s[0]); P(s[2])
Thread 1: V(s[1]); P(s[0]); V(s[1]); P(s[3])
Thread 2: V(s[2]); P(s[3]); V(s[2]); P(s[0])
Thread 3: V(s[3]); P(s[2]); V(s[3]); P(s[1])
```

C. What can go wrong with this proposed solution? Do **not** attempt to fix the problem; merely state what can go wrong.