

CS 385: HW2–Due March 16, 2004

Jon A. Solworth

March 9, 2004

1. Interleaving. Describe the possible outcomes in the following cases. Assume each variable to have a value 0 to begin with.

(a) $x \leftarrow 1$ $u \leftarrow 3$
 $y \leftarrow 2$ $v \leftarrow 4$

(b) $x \leftarrow a$ $u \leftarrow a$
 $y \leftarrow b$ $v \leftarrow b$

(c) $x \leftarrow 1$ $y \leftarrow 3$
 $y \leftarrow 2$ $x \leftarrow 4$

(d) $x \leftarrow u$ $u \leftarrow y$
 $y \leftarrow v$ $v \leftarrow x$

2. Interference. In each case explain if there exists any interference or not. If there is, clearly explain which ones are affected and how.

(a) If there exists a shared array `count[]`, the statement

`count[pid]++;`

where `pid` is the process id. The above statement is executed in each process.

(b) if `p==q`, then does `*p=5` interfere with `*q=3` ?

(c) Does `a[2*i]=3` interfere with `c=a[i]` ?

(d) Does the following two statements interfere with each other ?

`y=x/4`

`z=x/2`

3. Consider an atomic operation `compareAndSwap(*a, *b)`, which swaps the value pointed to by `a` and `b` if `a>b`. Show how a critical region can be implemented using this instead of `testAndSet()`.

4. In which of the following case(s) is `ptr` referenced as shared variable?

```

(a)      ptr = malloc(size(int));
(b)      int X() {
          char *ptr = "Hello World!!";
          }
(c)      char *ptr = "Hello World!!";
          int main() {
          printf("%s", ptr);
          }
(d)      int main()
          {
          int fd;
          char *ptr;

          fd = shm_open("/testShm", O_RDWR);

          ptr = mmap(NULL, PROT_READ|PROT_WRITE, MAP_SHARED,
                    fd, sizeof(int), 0);

          }

```

5. Semaphore.

(a) Using semaphores create an atomic section which does

```

transfer(i, j, amount)
{
    atomic {
        a[i] += amount;
        a[j] -= amount;
    }
}

```

(b) Alloc(i) and Free(i): An array of 10 elements. Write alloc() and free() functions to manage the use of these elements. Maintain the free nodes on a linked list.

(c) There are 4 atomic sections and 4 variable A, B, C and D shared amongst these 4 sections. Each section uses variables as follows:

Section 1	Section 2
A and B	B and C
Section 3	Section 4
C and D	D and A

Code up a solution that maximizes access to the variables while guarantying mutual exclusion.