CS 4760 **Operating Systems** Test 1
Name: Fall 2009 Max Pts: 49

Important: This is an open book test. You can use any books, notes, or paper, but not exchange anything with other students. You are not allowed to use any electronic/communication devices, including a calculator. *Do not log into the computer during the test. Switch off your cell phones.* Any calculations and rough work can be done on the back side of the test pages. You will lose five points for not writing your name.

1. [6 pt] Explain how you can change a program while keeping the same process.

2. [6 pt] One of the features to distinguish between `stdout` and `stderr` is that `stderr` is unbuffered while `stdout` is buffered. Can you change `stdout` to be unbuffered? How will you do that?

3. [2+2+5 pt] Consider a memory system with cache memory and main memory. The access time for cache memory (T_c) is 100ns while that for main memory (T_m) is 1200ns. The cost for cache memory (C_c) is 0.01 cents/bit while that for main memory is 0.001 cents/bit.

   (a) What is the cost of 1 Mbyte of main memory?

   (b) What is the cost of 1 MByte of main memory using cache memory technology?

   (c) If the effective access time is 10% greater than the cache access time, what is the hit ratio H?

4. [6 pt] What is the difference between an interrupt and a trap? How can you handle multiple interrupts?

5. [10 pt] Consider the following version of bakery algorithm for n processes:

```
extern int    number[n];                    // In shared memory

void process ( int i )     // i is process id
{
    while ( 1 )
    {
        number[i] = 1 + max ( number, n );
        for ( int j = 0; j < n; j++ )
            while ( number[j] && ( number[j], j ) < ( number[i], i ) );

        critical_section();

        number[i] = 0;

        remainder_section();
    }
}
```

Does this version violate mutual exclusion? Explain why or why not?

6. [6 pt] The prime motivation of going for higher level synchronization methods (monitors or condition critical regions) is to avoid the mistakes a user may make with the lower level methods (semaphores). Yet, almost all the higher level methods are based on using semaphores. How do you reconcile that?

7. [6 pt] Explain the difference between the wait operation in a semaphore and a condition variable.