CS 4760	**Operating Systems**	Test 1
Name:	Spring 2015	Max Pts: 44

Important: This is an open book test. You can use any books, notes, or paper but no electronic device. *Do not log into the computer during the test, or use any electronic or communications device. Change your cell phones to silent mode.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. You will lose five points for not writing your name.

1. [6 pt] In the resource manager view of an operating system, we suggested that the operating system must be fair but then, we contradicted it by adding priorities for different processes. How do you reconcile this contradiction?

2. [6 pt] Suppose a stack is to be used by the processor to manage procedure calls and returns. Can the program counter be eliminated by using the top of the stack as a program counter?

3. [6 pt] How can you change a process while keeping the same program? What command will you use to replace a program while keeping the same process?

4. [6 pt] What is the purpose of system calls? How do system calls relate to the OS and to the concept of dual-mode (kernel-mode and user-mode) operation?

5. [10 pt] Consider the following program:

```
P1 : {                                P2 : {
extern shared int x;                  extern shared int x;
x = 10;                               x = 10;
while ( 1 ) {                         while ( 1 ) {
    x = x - 1;                            x = x - 1;
    x = x + 1;                            x = x + 1;
    if ( x != 4 )                         if ( x != 4 )
        printf ( "x is %d", x );              printf ( "x is %d", x );
    }                                     }
}                                     }
```

Note that the scheduler in a uniprocessor system would implement pseudo parallel execution of these two concurrent processes by interleaving their instructions, without restriction on the order of the interleaving. Show a sequence (i.e., trace the sequence of interleavings of statements) such that the statement "`x is 4`" is printed.
Hint: You should remember that the increment/decrement at the source language level are not done atomically, that is, the assembly language code:

```
LD    R0,X    /* Load R0 from memory location x           */
INCR  R0      /* Increment R0                             */
STO   R0,X    /* Store the incremented value back in x    */
```

implements the single C increment instruction (`x = x + 1`).

6. [6 pt] Unix may be unsuitable for real-time applications because a process executing in the kernel mode may not be preempted. Elaborate.