CS 4760	**Operating Systems**	Test 1
Name:	Spring 2010	Max Pts: 43

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. [9 pt] Briefly describe the three techniques for I/O: programmed I/O, interrupt-driven I/O, and direct memory access.

2. [6 pt] An I/O-bound program is one that, if run alone, would spend more time waiting for I/O than using the processor. A processor-bound program is the opposite. Suppose a short-term scheduling algorithm favors those programs that have used little processor time in the recent past. Explain why this algorithm favors I/O-bound programsand yet does not permanently deny processor time to processor-bound programs.

3. [6 pt] What are the two modes of process execution? Why do we need both of them?

4. [6 pt] Explain the concepts of synchronous vs asynchronous communications in message passing. Give an example of each for both send and receive operations.

5. [10 pt] Consider the following program to resolve critical section problem for two processes:

```
extern boolean blocked[2] = { false, false };  // In shared memory
extern int     turn = 0;                       // In shared memory

void process ( int i )     // i is process id
{
    while ( 1 )
    {
        blocked[i] = true;
        while ( turn != i )
        {
            while ( blocked[1-i] );     // do nothing
            turn = i;
        }

        critical_section();

        blocked[i] = false;

        remainder_section();
    }
}
```

Does this version satisfy our protocol for the critical section problem? Explain your answer.

6. [6 pt] Describe in detail the steps taken by an operating system to handle an interrupt. Name the data structures that are needed to handle the interrupts.