CS 376	**Operating Systems**	Test 1
Name:	Winter 2002	Max Pts: 50

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. *Do not log into the computer during the test.* 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. [5 pt] A number of processors have instruction set which allows a test for accumulator being zero without a comparison preceding it. For example, look at the following assembly code:

```
LOAD        X               # Load the contents of address X in accumulator
BZ          LOC             # Branch to label LOC if accumulator is zero
```

   This is generally achieved through a side effect. Can you specify how is this achieved?

2. [6 pt] What is the difference between interrupts and traps? Give two examples of each.

3. [8 pt] Is it possible to have two system calls defined in Unix with the same name? Why will someone want to do that? If it can be done, how will you do it? If it cannot be done, why? so?

4. [8 pt] I just bought a superfast machine but it only has one CPU. The machine supports multithreading and has benchmarks to prove an improvement in performance. However, the benchmarks are reported in terms of number of processes per minute, with typical load. Should the vendor use another performance measure than the number of processes? Explain your answer.

5. [5 pt] What is the maximum number of processes in the system at any time using the following code segment:

```
extern char mypath[];
for ( int i ( 0 ); i < 10; i++ )
{
    pid_t pid, pid_out;
    unsigned char status;
    if ( pid = fork() )
pid_out = wait ( &status );
    else
execl ( mypath, "child", "parameter", NULL );
}
```

Assume that `child` performs some simple computation and returns the result, that is captured in `status`.

6. [8 pt] We went over the `test-and-set` instruction to solve the critical section problem with hardware support. However, in some machines, the `test-and-set` instruction is not available, but an equivalent is touted to be a `swap` instruction that atomically swaps the contents of two memory locations. Can you use `swap` to solve the critical section problem? How?

7. [6 pt] Consider the following code for P that was discussed in class. Notice that the number of P and V on corresponding semaphores do not match. Is there something wrong in this code then?

```
void P()
{
    mutex.P();
    if ( --count < 0 )
    {
        mutex.V();
        delay.P();
    }
    mutex.V();
}
```

8. [4 pt] Event counters are always incremented and never decremented. Can this cause a problem (theoretically)?