CS 4760	Operating Systems	Test 1
Name:	Fall 2003	Max Pts: 36

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.* 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] Distinguish between the extended machine and resource allocator views of an operating system. What provides the extended machine view in Unix and Windows?

2. [6 pt] Explain the difference between a thread, a primitive (as in system call), and a process.

3. [6 pt] Memory management emphasizes process isolation for the purpose of security. Yet, a number of applications require memory to be shared across processes (example, your project). How does Unix achieve a balance between the two? Does it put more responsibility on user in any way for resource management?

4. [6 pt] What is the difference between a daemon and a zombie in Unix? How does the presence of either one of them affect the performance of the system?

5. [6 pt] We discussed a solution for the dining philosophers problem by using monitors. Does it satisfy our protocol for interprocess communication? Explain your answer for each condition to be satisfied in the protocol.

6. [6 pt] Look at the following code for producer consumer problem using semaphores. Does it communicate effectively? Is there a problem? Identify it. Assume that the variables have been appropriately defined.

```
void producer()
                                          void consumer()
{
                                          {
    while (1)
                                              while (1)
    {
                                              {
        item i = produce();
                                                  mutex.P();
        mutex.P();
                                                  full.P();
        empty.P();
                                                  item i = remove();
        put ( i );
                                                  empty.V();
        full.V();
                                                  mutex.V();
                                                  consume ( i );
        mutex.V();
    }
                                              }
}
                                          }
```