CS 376	**Operating Systems**	Test 1
Name:		Maximum Points: 53

Important: This is an open book test. You can use any books, notes, or paper. *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.

1. [4pt] Set up your `PATH` variable so that it searches the following directories in order:

   (a) `/usr/ucb`

   (b) `/bin`

   (c) `/usr/sbin`

   (d) `/usr/bin`

   (e) `bin` directory of the user using this path

   (f) the working directory

   If there is a file called `whereis` in `/usr/sbin` and also one in your own `bin`, which one will be executed when you type `whereis` on the command line? (Assume you have execute permission for both of the files). If your `PATH` variable is not set to search the working directory, how can you execute a program located there?

2. [4 pt] What statement can you put in your `.cshrc` file to prevent yourself from accidentally overwriting a file when you redirect output? How can you override this feature? What statement on admiral/jinx disallows the use of `ctrl-d` to log out the current session? How do you fix that?

3. [10 pt] I just `ftp`'ed a bunch of files from a PC and they all turned out to be uppercase names. Write a shellscript (or set of commands) to change all the files with uppercase names to lowercase names. If two files have the same uppercase and lowercase name in the directory, make sure that the uppercase file name is translated to lowercase name with a `.u` extension added (do not add the extension unless required). Do not modify the contents of the files.

4. [10 pt] A main memory system consists of a number of memory modules attached to the system bus. When a write request is made, the bus is occupied for 100 nanoseconds (ns) by the data, address, and control signals. During the same 100 ns, and for 400 ns thereafter, the addressed memory module executes one cycle accepting and storing the data. The operation of the memory modules may overlap, but only one request can be on the bus at any time.

   (a) Assume that there are eight such modules connected to the bus. What is the maximum possible rate (in words per second) at which data can be stored?

   (b) Sketch a graph of the maximum write rate as a function of the module cycle time, assuming eight memory modules and a bus busy time of 100 ns.

5. [10 pt] Is the following program suitable for solving the critical section problem? Initially, `blocked` is assigned a value `{ 0, 0 }` while `turn` is assigned the value `0`. Explain your answer.

```
extern int blocked[2];   /* shared array */
extern int turn;         /* shared variable */

main()   /* process p0; similar code exists for process p1 */
{
#define ID 0             /* other process defines it as 1 */
    do
        blocked[ID] = 1;
        while ( turn != ID )     /* not my turn */
        {
            while ( blocked[1-ID] );
            turn = ID;
        }

        critical_section();

        blocked[ID] = 0;

        remainder_section();
    while ( 1 )
}
```

6. [10 pt] Assume that each of the five philosophers, i, in the dining philosophers problem executes the following segment of code:

```
void philosopher ( int i )
{
    do
        P ( mutex );
        P ( chopstick[i] );
        P ( chopstick[(i+1) % 5];
        V ( mutex );
        eat();
        V ( chopstick[i] );
        V ( chopstick[(i+1) % 5];
    while ( 1 );
}
```

Does this code satisfy all requirements of the dining philosophers problem?

7. [5pt] Why must the `join` operation in the `fork/join/quit` primitives be indivisible?