Homework 1

Due Date: October 12, 2000 **200 Points**

1. (*20 points*) Chapter 1, exercise 8
2. (*20 points*) Chapter 1, exercise 9
3. (*20 points*) Chapter 1, exercise 11
4. (*20 points*) Chapter 1, exercise 15
5. (*30 points*) *Robust Programming* handout, exercise 5
6. (*30 points*) *Robust Programming* handout, exercise 13
7. (*60 points*) This exercise asks you to look at a standard UNIX C library for problems with robustness. Please write three programs that use functions from the standard I/O library. You are to call the functions in such a way that they cause the library function to crash, or generate unpredictable results. Turn in the following:

   a. To demonstrate "crashing," use *gdb* output to show that the crash occured *within* the standard I/O library function. That is, the program must call to the standard I/O library function, but cannot return from that call.

   b. To demonstrate "unpredictable results," run your program (without changes) on at least two different types of computers in the CSIF (for example, once on a Linux system and once on an SGI) and show that the results of the function differ (you can use *gdb*, or print the relevant values).

   Please submit both the programs and typescripts for each program showing the crash or the unpredictable results. ***Important note:you must supply the correct type of argument for the functions. You may not, for example, pass a character pointer where a file pointer is expected.***

   When you submit your program, please place your programs, Makefile, README, and any other relevant files into a subdirectory. Please call the directory *stdio*. Generate a tar file called *stdio.tar.* Use *handin* to submit *stdio.tar.* We will un-tar it, run *make*, and go from there.