Homework 1

Due Date: January 18, 2006 Points: 100

1. (*14 points*) Text, problem 1.1.
2. (*15 points*) Text, problem 1.3.
3. (*21 points*) Text, problem 1.7.
4. (*10 points*) Text, problem 1.14.
5. (*20 points*) Handout, exercise 2.
6. (*20 points*) Consider the Linux system call *lseek*(2). It changes the position of the offset of the file descriptor, so reading or writing after the call begins at the position indicated by the second and third arguments to *lseek*. On success, the function returns the new offset. On failure, it returns -1. But -1 is also a valid offset, representing a position of 0xffffffff. The manual page suggests using *errno* to distinguish between success and failure. Please describe ***in detail*** how to do this, and give sample code.

Extra Credit

1. (*40 points*) This exercise asks you to look at the standard I/O library for problems with robustness. Please write two programs that use functions from the strings library (the functions in that library are listed in the *stdio*(3) manual page on the Linux CSIF systems). You are to call the functions in such a way that they cause the library function to crash.

   To show that your program crashes the standard I/O function, use *gdb*(1) output to show that the crash occured within the library function. That is, the program must call the standard I/O library function, but cannot return from that call.

   Please explain why the library function fails. Support your hypothesis with references to the *gdb* output (or to source code, if that is available—but state how you know your version of the source code is the version actually installed!)

   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. Your two programs must not use the same technique to cause the crashes. For example, if you discover a certain argument to *fread*(3) causes a crash, you cannot use that same argument to cause a crash in *fwrite*(3). However, if a different argument will cause *fwrite*(3) to crash, feel free to use that.