Homework #7 (30 points)

Part I (10 points)

Due, Monday, March 26

Look at the program stored in `/unc/brock/190/s90/home7/sesame.c` on `dopey` and write a couple of pages describing what it does.

Part II (20 points)

Due, Monday, April 2

The file `/unc/brock/190/s90/home7/sesame` is an executable produced by compiling the file `/unc/brock/190/s90/home7/sesame.c`. This executable has been `setuid` to the user `brock`. Write a C program that `execl`s this executable in such a way that it adds your user name to the file `/unc/brock/190/s90/home7/THE.list`.

Rules of engagement

Do part I by yourself. If you wish, you may do part II with a single partner as long as both people do about the same amount of work.

Warnings and Hints

This should be a difficult assignment. The amount of programming required is not that large (57 lines can be enough), but mastering the intricacies of interprocess communication will be a challenge.

Signals and signal handling are hard to use. You'll probably need to read several `man` pages before starting.

Your program must perform several time-critical operations, *e.g.,* creating a file within a few seconds. I've been generous in these time requirements, but at times the load on `dopey` may be so high that your process will not be scheduled in time to meet these requirements. Use the `uptime` command to see if the system load or number of users is high.

You may have to occasionally make your program `sleep` in order to make it work correctly.

The information you get back from `sesame` is sparse. When debugging, it would be a good idea to use a copy of `sesame` that you've modified to be more talkative. Similarly, you'd do well to test the return codes of system calls and library routines in your program.

Don't assume an `exec` is always successful!