Project 1: The *ish* Shell

Due: October 12, 2004

Implement the Unix shell described in the accompanying man page. Submit your solution (source files only) using the *turnin* command; use the assignment name *552ish*. Be sure to include all the files needed to make the shell. Your shell should compile by typing *make*, producing an executable file named *ish*. You will be penalized if your code contains unresolvable external references.

To help you get started, I suggest that you become familiar with the Unix system calls defined in Section 2 of the Unix programmer's manual. You should also be aware that there are several library routines (Section 3) that offer convenient interfaces to some of the more cryptic system calls. However, you may not use the library routine *system*, nor any of the routines prohibited in the *ish* man page. Chapters 8, 9, and parts of 10 in the Stevens book are also very relevant.

You may, if you desire, use the *lex* and *yacc* programs to implement your command parsing. Even if you haven't used these programs before, you will probably find it much simpler to learn them than to write your own parser. I will provide the lex and yacc input files *ish.l* and *ish.y*, respectively.

To implement *ish* you will obviously be creating a process that forks off other processes, which may in turn fork processes, etc. Unix limits the number of processes you can have; if you run out you won't be able to do very much so be careful. While you are testing you may want to somehow limit the number of shells you can have running at once.

When in doubt about the functionality of *ish*, or how it should behave in a particular situation, model its behavior after that of *csh*. Questions of general interested should be posted to the *cs.course552* newsgroup.