

CS 2240, Fall 2015,
Assignment 5 – Write a Shell
Due November 10

You are to program your own shell. The minimum functionality is that it must

- a) Execute an arbitrary number of piped commands on a single command line
- b) Handle redirection for both first and last command.
- c) Have two built in commands: `umask` and `cd`
- d) Allow backgrounding (a final `&` which causes the shell to

You have been provided a good number of examples and explanation of the key system calls: `fork()`, `exec()`, `dup2()` and `pipe()`, as well as instances of the `wait()` family of calls. You have also been provided `makeargv()` which takes care of tokenizing the command line into a form ready for `execvp()`.

This assignment is a kind of rite of passage, perhaps only because the thinking and coding is somewhat different since `fork()` causes you to look at a single piece of source with two different viewpoints. The amount of code proves to not be excessive. The examples you have seen should actually make things relatively understandable since process communication and structure is what we are really thinking about here.

Some tips:

- You will do well to do some planning on overall organization of the pieces you have seen. An orderly attack as you add functionality will help. In particular, thinking through how an arbitrary number of pipes can be implemented amuses me, though Scott finds that another approach is more palatable. I will discuss.
- Have some toy programs to work with. You have seen a little thing that upper cases the stream of characters already. It is useful for testing to have 2 or 3 pipeworthy programs at the ready.
- `gdb` can be used and when forking either `'follow'` the child or stay with the parent.
- A command file that can be redirected into your shell can save key strokes. On each line of the file is what the user (you would otherwise type).

Special Extra Bonus Feature (SEBF)

Add something kewl. Since James is an industrious fellow and hangs around Scott he has already implemented `'history'` which is a nice feature. A `'tee'` capability exists in some shells. Handling the keyboard can be a useful digression for adding control characters.