ELEC 377 Lab 2 Documentation

Gia Lee, Valerie So, William Robson

Input :

All interaction with the shell is done via the command line. The string of characters is stored in the commandBuffer and sent to splitCommandLine to be processed into something the program can use.

SplitCommandLine and its helper function skipChar take the input string and split it up into arguments, using the start and end of the function as well as any spaces in the string to figure out where the arguments are. The number of found arguments is returned to nargs, and the args[] array is filled with the arguments in the order they were found in the commandBuffer.

There is a lot of edge cases here for the function to catch, mainly weird groups of spaces in the line.

Internal Commands:

If at least one argument is put in the command line, then the program will first try to use it in doInternalCommand function.

doInternalCommand checks the first argument against a list of defined internal commands. If a command with a matching name is found, a function call to that command is made and passed any arguments it may need. These are also “homemade” functions There are 4 implemented commands, exit, pwd, ls, and cd that have the same basic functionality as any other shell.

If a command is not on that list, then it is passed to another function that tries to execute an external command.

External Commands:

DoProgram is the function responsible for trying to run external commands.

It takes the arguments and checks through the path for any matching commands. If one is found, it will build a proper path to it, fork a child process, and use the special built in `execv(commandPath, args);` function with the path and arguments to execute the function,