CS 4375

Professor Ward

**MicroShell Assignment**

Purpose 1: Gain experience weaving system calls into a functional program.

Purpose 2: Learn to manipulate the properties of processes.

Goal: Build a shell to support command-line use of POSIX system calls.

Functionality

1. Read a unix command from the user, execute it, and repeat. Handle at least the basic commands (**ls**, **cat**, **grep**, etc., typically found in **/usr/bin**), with all their normal parameters (options and arguments). [5 points]
2. Terminate if the input is **quit**. [1 point]
3. Output an inspiring phrase when the command is **inspiration**. The phrase to print is determined by the value of the environment variable **phrase**. [1 pt]
4. Support background tasks, that is, tasks which run without requiring the user to wait before the next command, specified with **&**. [3 pts]
5. Accept commands from a file if one is specified; ignore lines starting with **#** [1 pt]
6. Also run commands that name a program anywhere in the path. [3 points]
7. Change directories with the **cd** command. [1 pt]
8. Support redirection of output with **>** . [2 pts]
9. Support redirection of input with **<** . [1 pt]
10. Support simple pipes with **|** . [4 pts]
11. Print informative error messages: [3 pts]
    1. If a command is not found, print “command not found”.
    2. If a file whose name matches the command is found, but that file is not executable, print “not executable”.
    3. If a command fails (has a non-zero exit value *n*), print "Program terminated: exit code *n*."

Other factors: 5 points for general code quality and report quality.

Points possible: 30. Aim for 25 if you want an A, 20 for a B, and 15 for a C.

Constraints:

Use only the following python libraries:

* os
* sys
* re

From os, use only:

* pipe()
* fork()
* dup() or dup2()
* execv() or execve()
* wait()
* open() or create() and close()
* read() and write()
* chdir()
* environ()

If you think you need to use another system call, consult with the instructor or TA. In any case, do not use system(), which(), execl**p**(), execv**p**(), or execv**p**e().

Submit

1. a report including
   1. snapshots showing that your system works from the console: (functions a ~ d)
   2. output showing that your system works in batch mode (function e)
   3. evidence that your system handles functions e ~ l. We will later provide test code to help you do this.
   4. a paragraph or two that describe: interesting features of your system, where you had trouble, and/or what you learned.
   5. your code
2. Have your python code in good shape and ready to give a live demo in class, or to provide it on request for us to test.

Due February 19

Hints

1. For the basic loop, you can borrow from anyone’s chatbot code.
2. For each command, create a child process that uses **execv** to run it with its parameters.
3. Borrow code from Dr. Freudenthal’s demos in **https://github.com/robustUTEP/os-shell**
4. Read [**http://www.rozmichelle.com/pipes-forks-dups/**](http://www.rozmichelle.com/pipes-forks-dups/)
5. “Believe in yourself," "Never give up," "You can do anything you set your mind to," "Dream big," "Embrace the journey," "The only limit is the one you set," "Make a difference," "Be the change you want to see," "Follow your heart," "The best is yet to come," "Every step counts," "Don't let setbacks define you," "You are capable of more than you think." "Keep pushing," "One step at a time," "Tough times don't last, but tough people do." "You are enough," "Trust your instincts," "Own your power."  "Focus on the good," "Think positive," "Every cloud has a silver lining." "Just do it," "Don't wait for the perfect moment," "Make it happen."