COSC 6360-Operating Systems
Assignment #1: A Simple Shell
Due Date: October 10 at 11:59:59 PM.

OBJECTIVE

This assignment should help you to understand the functions of a command language interpreter and learn how to create processes and do input/output redirection under UNIX.

SPECIFICATIONS

Your program should prompt for input from its standard input. (Use the prompt "`MyShell:`" to avoid confusions.) Every line of input will represent a single command. It will consist of a sequence of arguments separated by either an arbitrary number of spaces or one of the two I/O redirection characters ("**<**" and "**>**")). The first of these arguments will normally be the name of a program to be executed. All other arguments are to be passed to that program. An error message should be returned every time that an inaccessible program file has been specified in the input line.

Your shell should be able to handle background processes and input/output redirection:

- ***Background processes:*** Any process whose command line starts with the keyword "**bg**" should be executed in the background; that is, its standard input and standard output should be *closed* and the shell should not wait for its completion.
- ***Input/output Redirection****:* If a command line includes a "**>**" the next argument represents the name of a file to which the standard output of the command should be redirected as in:

  ```
  ls > listing
  ```

  Similarly, if the line includes a "**<**" the next argument represents the name of a file from which the standard input of the command should be read as in:

  ```
  wc -w<file
  ```

In addition, your shell should handle internally two process control commands:

1. **`processes`**, which displays a list of the background processes (process id and original command line) that have been initiated by the shell and are not yet terminated, and
2. **`exit`**, which kills all background processes your shell has forked has forked and terminates it.

NOTES

1. With the exception of "**>**" and "**<**", your program should not treat the special characters and meta-characters of **`/bin/sh`** and **`/bin/csh`** in any special way. So you could, if you ***really*** wanted it, create files named "***&**" and other illegal Linux names.
2. There is no need to worry about signals.
3. Each time, your shell has to execute a command that is not built-in and does not have a full pathname, it should search though the list of paths in the environment variable **PATH**.
4. You are not allowed to use **`system()`** or **`execvp()`**.

HINTS

1. Before you start your assignment, familiarize yourself with the UNIX functions **access()**, **`fork()`**, **`wait()`**, **`execv()`**, **`dup()`**, **`close()`**, **`exit()`**, **`_exit()`**, and **`kill()`**.
2. You will find that the command **`cd`** will not work properly (but the fix is very simple).
3. Remember that UNIX functions normally return a zero value when they are successful.

WRITING YOUR PROGRAM ON YOU COMPUTER

The best way of experimenting the full power of Linux is to develop your program on a Linux machine, virtual or real. When this is not possible, there are easier alternatives:

- If you have a Mac, there is nothing to install, as the macOS command line interface is a Unix/Linux shell.
- If you have a PC with an up to date 64-bit version of Windows 10, you **should** install a Linux shell. The best guide on how to do it comes straight from Microsoft:

  https://msdn.microsoft.com/en-us/commandline/wsl/install_guide
- If your computer runs any other version of Windows, your best bet is to install Cygwin from **cygwin.com**. The process is very straightforward, the tool very well integrated with other Windows apps, but not all the functions of Linux are always well emulated.

SUBMISSION PROCEDURE

Once you have submitted your assignment though Blackboard, you must schedule an appointment with the TA to get a chance to explain how your program works. ***This is the only way you will get your assignment graded.***

This document was updated last on Friday, September 14, 2018.