CSE303 Programming assignment 2. **Spring 2020**

The Shell or Command Line Interpreter is the fundamental user interface to an operating system (OS). Your second project is to write a simple shell (in C) that has the following properties:

1. The shell must support the following internal commands:

a. clr - Clear the screen.
b. dir <directory> - List the contents of directory <directory>.
c. environ - List all the environment strings.
d. run prog p1 p2 … exec's to prog passing the parameters p1 p2…
e. sleep <seconds> child process waits for <seconds> seconds.
f. quit - Quit the shell.

2. All other command line input is interpreted as program invocation, which should be done by the UNIX system() function.

3. Your shell should implement each command by forking a new process and overlaying the child process with the requested function. The parent process should wait for the command to finish before continuing unless the command is followed up with a "&" character.

Examples:

wop219 >> dir . {lists directory contents of the . (current) directory, shell waits for the command to complete.
wop219 >> environ & { lists environment variables, shell does not wait }

4. The command line prompt must contain your Lehigh email username. Example:
wop219 >>

Project Requirements

1. Design a simple command line shell that satisfies the above criteria and implement it on the alpha.fdu.edu UNIX platform.

2. The source code **MUST** be extensively commented and appropriately structured to allow your peers to understand and easily maintain the code. Properly commented and laid out code is much easier to interpret, and it is in your interests to ensure that the person marking your project is able to understand your coding without having to perform mental gymnastics!

3. The submission should contain only 1 source code file. The person marking your project will be building your shell program from the source code provided. If the submitted code does not compile it cannot be marked!

Submission

All that is required is your source code (your .c file). The file should be named <yourname>.c (not a doc file, txt file, or anything but a .c file). The .c file should compile using the C compiler in your docker container AND on the sunlab machines. As you design your shell, keep in mind we may want to add commands to it to interface with programs or utilities we create and install into the OS we build.

I am providing a sample shell that implements a couple of commands but not in accordance with anything related to this project. It does, however, use MOST of the Application Program(ming) Interface (API) calls you'll need to accomplish this project.