Operating Systems & Adv. Mobile project

HW1: simple Shell (SiSH)

Due by Sep. 28

Introduction

Shell is a small program that allows a user to directly interact with your operating system. A user can command (give orders) to run program through shell. To do that, shell takes input string from a user, and make it run. When the specified program completes its execution, the shell takes another input to run another program.

In this homework (programming assignment), you will have to write a small shell program.

Shell usually works with low-level OS functions such as system calls. You may look into detailed usage of the following system calls: fork, execve, wait. Fork is a system call that creates another user process; it actually creates a copy of the caller. Then, the process becomes two different processes, return with different values. For parent process (that creates the child), fork() call returns with some number, that is the process id of the child process. For child process, fork() call returns with zero. Note that both (parent and the child) uses the same code base.

Execve is another important OS function that change the process into another process. The call takes the filename as argument, that the user wants to execute. The call discards the current program context (address space, changed memory values, register values, etc.), and loads the specified program into the memory. It then begins execution from the very beginning point of the program. Now that you can run another program that you want to execute in the child process.

Finally, wait() is a system call that stops the execution of the parent process. Parent process has to wait until the child completes its execution. The wait() call implements such waiting.

You may further look into the following APIs: strtok_r, getenv.

Specific requirements

1. make & compile: write a makefile that compiles your code. I will just type make on the terminal.
2. Start of SiSH: by entering the executable filename, your shell starts.

3. End of Sish: SiSH finishes execution when it gets ‘quit’ string from the user.
4. Operation of SiSH:
    1. Input: takes program name as input string
    2. Execution: it has to execute every single executable program in the filesystem, if it has proper privilege
    3. Execution path: to simplify (contract) the filename (full path beginning with ‘/’), SiSH should look into directories, in PATH environment variable.
    4. PATH environment variable holds the ‘:’-separated string, that specifies multiple locations in the filesystem.
    5. During the execution of the user-input program, shell should not be active.
    6. Repetition: When the given program completes its execution, it receives the next input string, to run another program.
5. You can specify different shell prompt using getenv function. (e.g. your current working directory (PWD, TIME, USER, etc.)
6. You can take additional input parameters for the executing program, and pass them to the created process. Please find manual pages of execve system call.
7. Document:
    1. You should include some report for your code project, hw, etc.
    2. The report should include general/brief introduction to your program.
    3. The report should include specific instruction to make your program.
    4. The report should include specific working example. (screen capture, whatever)
    5. The report can include your personal ideas, feedback messages to me.
8. Submission
    1. You have five freedays. (for all projects, homework assignments)
    2. Submit your code through github account, in your own branch.
    3. Branch should be your student ID.
    4. Test before push. Do not push the broken code in git.
9. Evaluation
    1. It’s better if your code is different from the others.
    2. More functionalities, more error handling is better.

No restrictions on working environment, if it supports POSIX standard programming interface. However, I strongly recommend you to stick to assam server; you may have different working environment, according to the running platform (OS).

No copy allowed. Please push your code in github hw1, in different branches. Do not push your code to master branch! Note that you can look at others’ code, and vice versa.

Different code implies better evaluation.

Enjoy your system-level programming!