Programming Assignment 1

CSC456 Operating Systems

Fall 2021

Assigned Data: 9/3/2021

Due Date: 11:59 PM, 9/13/2021

1. Objectives

- Hands-on experience of using system calls related to process management
- Improved understanding of process management
- Implementation of a simple command line interpreter

2. Overview

Students will implement a simple shell (a command-line interpreter). It will read user commands (i.e., any Linux/Unix commands such as *ls*, *cp*, *mkdir*, …) from Standard Input and execute commands using the system calls related to process management.

3. Requirements

- Your shell must display a prompt consisting of your last name. For example,

  Your last name> _

- Your shell must read a user command and its argument(s). Your shell must be able to take any number of arguments. For example,

  Your last name> ls -l ⏎

  Your last name> cp fileA fileB ⏎

- In the first example, there is one argument, '-l', and in the second example, there are two arguments, which are 'fileA' and 'fileB'.
-

Note that your program will be evaluated with random commands with one, two, and three arguments.

- Your shell must continue to take another user command after completing the execution of the previous command. For example,

  Your last name> ls

  . .. fileA.txt fileB.txt

  Your last name> _

- When a user types 'exit,' your shell must terminate. For example,

  Your last name> exit

  Bye!

- Your shell (a parent process) must wait until its child process (a user command) finishes its execution.

- To run a user command, consider using the system call 'exec()'. Read the manual of this system call (e.g., https://man7.org/linux/man-pages/man3/exec.3.html). You will find that there are different ways of using this system call. It is your task to read the manual and select an appropriate one for this programming assignment. For example, your exec() system call is used to execute a program with an arbitrary number of arguments.

- Your shell must run on a POSIX-compliant operating system.

4. Project Management Tool Requirements (1/4)

- Select a project management tool of your preference, e.g., SVN, Git, etc.
- Read the manual of your preferred project management tool
- Be prepared! You will use a project management tool for upcoming programming assignments.

5. What to turn in

Submit a zip file (**filename:firstname_lastname.zip**) that contains the following files:

- Your source and header(if you used one) files (e.g., 'c' or 'cpp', and 'h' files)
- A makefile (if you used one0
- All other files that are needed to run your program
- **A report** that describes briefly about how to run your program.
- Submit your zip file to **the dropbox on D2L.**

6. Evaluation Criteria

Your project will be evaluated based on the following:

Documentation 20% - your report clearly describes how to run your program and how you addressed the program requirements.

Compilation 15% - your code can be compiled without errors or warnings.

Correctness 60% - your program satisfies all the program requirements.

Readability and Misc. 5% - your source code must be well commented. Your submitted zip filename complies with the instructions in Section 4. Your zip file contains all files that are needed to run your program.