Operating Systems - Monsoon 2022

Arani Bhattacharya, Sambuddho Chakravarty

October 2, 2022

Assignment 1 (Total points: 70)

Due date: Oct 14, 2022. Time: 23:59 Hrs.

Basic Linux/Unix Shell

Linux (and other Unix like OSes), have "shells" or programs which present a command line interface to users to type commands in. In this assignment you need to use standard C libraries, including Linux system calls such as `fork()`,`exec()` family system calls and `wait()` family of system calls.

There are two kinds of commands – "internal" and "external". Internal commands are those which are interpreted by the shell program itself, without requiring a different program to handle the expected operations (of the said command). Examples of internal commands are like `'cd'`, `'pwd'`, `'exit'` *etc.* External commands on the other hand relate to commands which are not handled directly by the shell program but by an external program. Common examples include `'ls'`, `'cat'`, `'grep'` *etc.*

In this assignment, you have two tasks:

1. You need to design a simple shell that can handle **three**, internal commands – `'cd', 'echo'` and `'pwd'`. These commands would be handled directly by the shell. Your shell should also be able to handle **five** external commands – `'ls', 'cat', 'date', 'rm'` and `'mkdir'`. For these external commands you need to write individual programs to handle these commands. To handle these external commands, the shell should typically create a new process, using the `fork()` system 'call and within each process you need to use the `execl()` family system call to run the individual program. The parent program must also wait for the child program to terminate using the `wait()` family of system calls.

   For each of these commands, you need not handle all the command line options. Two options per command is sufficient. You need to document which two options you are handling and need to demonstrate correct functioning of the command with respect to (atleast) your chosen options. You also need to handle corner cases such as invalid options (graceful degradation).

2. The second task would be the achieve the above functionality of the shell using `pthread_create()` (instead of `fork()`) and `system()` (instead of

`execl()` family of functions). The thread based execution would be performed if the command is followed by the characters, "&t". The rest of the functionalities should remain the same. **Note:** you only need one set of external command programs which could be used with either versions of the shell, be it the that uses `fork()/execl()` or the one that uses `pthread_create()/system()`.

What To Submit

- The C program sources.
- `Makefile` to compile the source and generate the running binary for the shell.
- Write-up describing the system, the options the shell commands can take, the errors you handled and the assumptions you made. Also provide test cases to test the functioning of the shell.

Grading Rubric

- Successful compilation using Makefile – 5 points.
- Use of system calls like – `fork()`, `execl()` family of system calls, `wait()` family of system calls to handle external commands – 10 points.
- Use of API function calls – `system()` and POSIX Pthread family of functions (`pthread_create()`) to handle external commands – 10 points.
- Correct handling of commands (two options per command, as described earlier) – 20 points.
- Successfully handling atleast two corner cases for each of the commands – 20 points (List the bugs/errors/attacks that you defend against).
- Description of the systems, commands to execute and test the program and the assumptions that you made – 5 points.

Late Submission Policy

- Submitted on or before Oct 14, 2022 (23:59 hrs) – No points deducted.
- Submitted after Oct 14, 2022, but on or before Oct 25, 2022 (23:59 hrs) – 5 points deducted.
- Submitted after Oct 25, 2022, but on or before Oct 30, 2022 (23:59 hrs) – 10 points deducted.
- Submitted after Oct 30, 2022 – no points shall be awarded.