CSE 344 System Programming

Homework #2

Due April 13

In this homework, you are expected to develop a terminal emulator capable of handling up to 20 shell commands in a single line, without using the **"system()"** function from the standard C library. Instead, you should utilize the **"fork()"**, **"execl()"**, **"wait()"**, and **"exit()"** functions.

Your terminal emulator should include the following features:

- Each shell command should be executed via a newly created child process, meaning that multiple commands will result in multiple child processes.
- Proper handling of pipes ("|") and redirections ("<", ">") by redirecting the appropriate file descriptors.
- Usage information should be printed if the program is not called properly.
- Error messages and signals that occur during execution should be printed, and the program should return to the prompt to receive new commands.
- Aside from a SIGKILL (which also should be handled properly) the program must wait for ":q" to finalize its execution.
- Upon completion, all pids of child processes with their corresponding commands should be logged in a separate file. Each execution should create a new log file with a name corresponding to the current timestamp.

Make sure to test your program with multiple shell commands in **/bin/sh** to ensure its functionality.

Grading:

1) Compilation errors: -100 points.
2) Improper use of fork(), execl(), wait(), and exit() will result in a deduction of 100
3) Make sure to include a makefile with the "make clean" command. Failure to do so will result in a deduction of 30 points.
4) Your report is essential, as it will serve as the primary basis for evaluation. Failure to submit your PDF report will result in a deduction of 100 points.
5) Late submissions will not be accepted.
6) Failure to clean up after child processes or leaving zombies will result in a deduction of 30 points.
7) The presence of memory leaks (which will be verified with valgrind) will result in a deduction of 30 points.

Submission rules:

1) Your source files, your makefile and a report; place them all in a directory named "lastname_firstname_studentnumber", and zip the directory.
2) Your makefile should only compile the program, not run it!
3) Your report must be in PDF format and should include a detailed explanation of how you solved the problem, which requirements you met, and which commands you tested
4) You should do your homework on your own. Your homework will be compared against online sources as well as each other's homework. Proven cases of plagiarism will result in −100 grade.