CMPUT 379 Assignment 1

Chengxuan Li

2022-02-02

1 Objectives

This programming assignment provides some hand on experience in using Unix system calls for accessing and utilizing system time values, process environment variables, process resource limits, and process management functions. Additionally, this assignment provides a brief idea of the underlying structure and mechanism of a shell program that can interact and communicate between user and OS.

2 Design Overview

3 Project Status

The current status of the project is that the simple shell program meet all the program specifications mention in the assignment descriptions. However, it is possible for some edge cases to cause the shell program to behave differently or produce unexpected results. Also, the implementation in current status of the project is not yet optimal although it behave as what the specification describes. One of the reasons is that some programming concepts and strategies as well as refactor techniques are difficult to apply using C language.

For cdir command, if there are environmental variables presented in the path name, cdir command will try to replace all the environmental variables with its value if it is possible. Otherwise, it will simply ignore the '$' in path name.

For stop command, if it tries to stop a zombie process, the program usually complains that it cannot stop that process. That zombie process will most likely be removed by the system after SIGSTOP is sent. Consequently, lstasks command will still list that zombie process although it is removed by the system, and terminate command will complain when calling kill() with SIGKILL.

For run command, if there is no specified program name provided, it will only fork a process that will immediately return 0 from main function and become a zombie process.

For check command, if a process has too many child processes, check command cannot display all its child processes. The limit for the number of child processes is set to 132.

There are no specific signal handlers installed at the current status of the project for signal such SIGQUIT, SIGINT, etc.

4 Testing and Results

To tested my implementation, the simple shell program was tested on two different machines: lab machine and ManjaroI3, which is based on Arch Linux.

5 Acknowledgements

List of sources of assistance:

- Stevens, S. Rago, Advanced Programming in the Unix Environment, 3/E, Addison Wesley, 2013
- The Open Group Base Specifications Issue 7, 2018 edition IEEE Std 1003.1™-2017 (Revision of IEEE Std 1003.1-2008)
- Linux Man Pages