CMPUT 379 Assignment 1: Process Management Programs

Objectives

This programming assignment provided us with some hands on experience in using UNIX system calls for accessing and utilizing system time values, process environment variables, process resource limits and other management functions like fork(), waitpid(), and execl().

Design Overview

- The simple shell program was split into 3 modules where, all functions and prototypes related to the commands are in commands.c & commands.h, the main function and the function that parses user input with a global variable that tracks the number of accepted tasks in msh379.c & msh379.h and finally, a util.c & util.h providing general use functions like splitting a string into tokens, printing times.
- Includes error checking after important function calls.

Project Status

The project meets all the specifications of the assignment. I found programming in C without the help of string functions in C++ to be a tedious process. The program is expected to give us the expected output for all the commands but can also be broken/expected to behave differently for certain edge cases.

Initially, for the run command, I had trouble following the child processes where the parent goes to completion before the child process, this fixed with the help of a waitpid function in the parent process. This command also considers a task accepted if the fork() was successful, meaning a failed execlp would be counted as an accepted task, so one must be sure that their command would run successfully before using 'run'

For the check command, I faced some trouble trying to print the descendants of the target_pid , but was resolved with a struct array storing information of parent and their respective child.

Testing and Results

The testing of the program was done on the lab machine by coding on VS Code and using SCP to transfer the files onto a lab machine. The commands were tested using a myclock, mMyclock scripts and x11 programs xeyes and xclock, the outputs of the commands on these scripts closely matched those provided on eClass.

Acknowledgments

- Stevens, S. Rago, Advanced Programming in the Unix Environment, 3/E, Addison Wesley, 2013
- Linux Man Pages
- Various stack overflow threads