CMPUT 379 - Assignment 1

Objectives

The objective of this assignment can provide hands-on experience in using basic Linux system API, such as, process management. The assignment provides insight to process lifecycle, practice how to suspend or terminate a process by sending `signal` to a specific process. Also, learning IPC by creating a pipe to another process to read from its `STDOUT` or `STDERR`.

Design Overview

- Each command has its own handler function, keep code modular
- Keep the code as dry as possible by seperating repeated code into small helper function

Project Status

All required features are implemented with error handling for most important functions, however, some edge cases may not be handled properly.

I found writing code in C/C++ most challenging, it takes time to familiarize.

Testing and Results

Have three terminal window opened in advance.

- run `a1jobs` in the first terminal
- run `a1mon <a1jobs_pid>` in the third terminal
- type `run xclock`
- type `list`
- run `ps` in another terminal to check the both parant and child process are running, also compare the `PID` of child process with `ps` output, make sure they are identical
- check `a1mon` is showing both parent and child processes
- type `suspend 0`
- run `ps` in anoher terminal, check `xclock` state has changed to `T`
- type `resume 0`
- run `ps` in another terminal, check `xclock` state has changed to `S`
- type `terminate 0`
- run `ps`, check `xclock` state has changed
- type `exit`
- run `ps`, check if all parent and child processes no longer exist
- go back to `a1mon` check if it show target process was terminated

Acknowledgments

- APUE
- https://linux.die.net/man/
- https://stackoverflow.com/a/5907076/4557739