CIS 415 Operating Systems

Project <1> Report Collection

Submitted to:

Prof. Allen Malony

Author:

<Tong Guan>

<UO ID - tongg>

<Duck ID 951871617>

Report

Introduction

This is a basic Linux command-line shell project implemented using C language and system calls. Users can type common commands in a custom shell, like ‘ls’, ‘cd’, ‘pwd’…, and interactively react with the file system. What makes this project particularly interesting is its dual functionality. It includes interactive user mode and file mode. In file mode, our shell can automatically execute commands from a file and output the results to 'output.txt.

Background & Implementation

The implementation process of this project is akin to a tree with a hierarchical and interactive structure. At the beginning, it need to check whether it's in interactive mode or file mode based on user input. In interactive mode, it use 'getline' to accept user input. In file mode, the shell read the input file while directing the output to 'output.txt.' What's interesting is that, after receiving the commands, I handle them using the same logic.

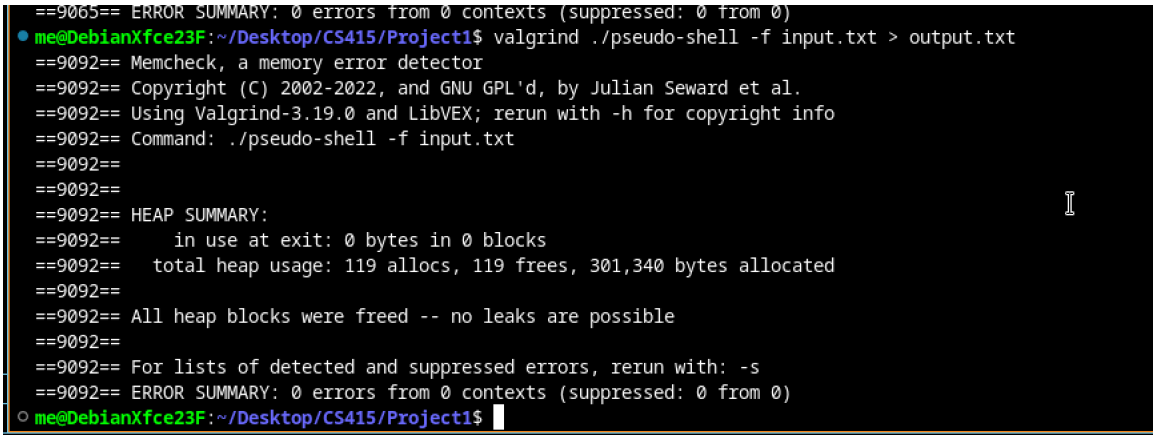
First, the shell need to parse these commands via tokens . Considering the various formats in which users may input commands, I need to parse two layers on the line: first by semicolon (';') and then by space (' '). After parsed the respective command and its associated file, it then find the corresponding execution function for these commands.

I find the 'cp' command the most challenging in this regard because, for the 'destinationfile,' it accepts three modes: 'file,' 'dir/file,' and 'dir.' For 'dir,' I need to append the source file to it to obtain the complete relative path.

This project also requires detailed and comprehensive error handling. At each step, I must check whether the command is correct, whether the file count is accurate, whether files can be opened, and whether file reading is successful, among other checks.

Performance Results and Discussion

My project implements all the features without any memory leaks, while performing well in both interactive mode and file mode.(here is a screenshot of file mode, the input file is downloaded from assignment)



Conclusion

In this project, I've learned a lot. The first lesson is how to build an entire system step by step using different files, especially in a structure with various branches. The second lesson is about testing, and I've come to realize the necessity of good and effective testing. To test my command execution functions, I designed many different 'input.txt' files myself. The debugging process can be frustrating, but I made an effort to stay calm through thorough testing and worked hard to modify the program.