Hari Kifle

Isaiah Britto

CS 4310.02 Operating Systems

Assignment 4: Report

Introduction:

In this assignment, we were tasked on implementing your own shell in C. The shell was limited to handling basic commands and needed to be able to execute any of them no matter the case. We had to implement all the basics including mkdir, rmdir, and cp, as well as being able to run an executable. This project was tough and it had its ups and downs, but the end result came out great.

Design:

The design revolves around a big if-else loop that has all the commands in it. When run, it starts a while loop that runs the system. It starts reading the user input. The input then is sent into a parsing method and is put into an array. After reading and parsing the string, it enters into the command method where there is a big if-else loop that handles each of the commands. It compares it to an array in the method that has all the instructions in it, and assigns a particular number to each, which it used to enter the if loop and execute. For commands that require different variables like the copy command which needed two pointers, we instantiated them in their respective loops. This design is simple, easy to read and understand, but it came with its own issues and errors that we needed to debug step by step.

Techniques:

Overall, when my teammate and I started, the first thing we did some research on how a shell was supposed to work. We looked at some of the basic commands and the arguments that needed to be passed in as well as the expected results of each. From there we made a general plan of who would get what done and went to work from there. To implement the commands, we used a list that would be searched through after to match the command argument with the desired functionality. Once the command executes this method returns the program back to its overall control loop where it awaits further user input in the shell for its next command. Most of the needed functions were in the C++ filesystem class so we just had to make the inputs compatible and account for any errors with the appropriate message.

Findings:

Overall, we ran into more problems than we expected. Many of them had to do with nuances of C++ as we both are not too familiar with the language overall. A lot of time was spent on trying to figure out how to pass arguments around and deal with single and double pointers. Overall, we both feel as though our knowledge and comfort level in C++ increased.

Conclusion:

After making this Shell with C++, we have a greater appreciation for the built-in shells of modern operating systems. This was a good amount of work just to get these basic commands implemented, and there’s a lot to expand on and add to if we worked on it more. This only scratches the surface of what Shells run in a modern operating system in order to handle all the functionalities of a computer but without the graphical user interface.