CIS 415 Operating Systems

Assignment <1> Report Collection

Submitted to:

Prof. Allen Malony

Author:

<Jacob Rammer>

Report

Introduction

To keep this short, essentially the purpose of this project is for us to create our own common Unix commands such as ls, pwd, mv, etc, using systems calls. In addition to systems calls in command.c we, or at least I, was introduced to some very useful functions I did not knew existed before. For example, strtok_r and strrchr are both functions I have never used before, and boy were they helpful. In addition to system calls, I believe this project was meant to help us with our file I/O handling, which it did.

I had to turn in this assignment late due to my VM crashing. It was both a good and bad thing. The bad thing was I lost a days' worth of work and so angry that night I did not eat. But on the flip side I had extra time while stewing to better develop my main.c file to something that isn't (I think) complete garbage.

Background

Since we were told in the beginning of class to essentially Google our questions, that's what I did. When creating a Unix command function such as ls, I Googled something along the lines of "c system calls list files". Since the manpage is huge and doesn't have the description of what every function does until you open them, this was a huge help. Once I had an idea of which system call to use, I looked at the manpage and implemented my system calls that way. During my research stage, https://www.tutorialspoint.com/ gave me a lot of good introductory info. Essentially, after I tested something to found it working, I implemented it and went on from there.

I thought that I would have to a lot more micro-managing that I did. I figured I would have to cd to the correct directory pretty much any time I wanted to do that. I was surprised to find out that C does a good job keeping track of that for you. One thing I am not happy about in my assignment is my numerous repeated for loops to remove the newline characters. If my VM did not crash, I would have condensed them into a function like I did in main.

Implementation

I'll start off with the problems first. My main issue was dealing with incomplete paths such as "../"or ".". I believe I do have all of that straightened out now and everything should work as it should. Going back to one of my points above, I was not happy with my repeated for loops to remove special characters. As far I know, there is not a way to remove specific characters from strings in C, so this is why I went this way. I think the way I handled my main is pretty nifty. I had two save pointers for strtok_r, one for the whole line, and another for each specific command. When checking for unexpected commands I checked the length of the command save pointer, which should be empty if the command is used correctly. As far as everything in command.c, I think they're pretty straight forward since it's just a lot of open / reads / writes, and directory creation.

```
else if(strcmp(command, "cp") == 0)
    {
        // cp needs 2 args
        tempArgument1 = strtok_r(NULL, "
", &commandPtr);
        tempArgument2 = strtok_r(NULL, "
", &commandPtr);
        if(strlen(commandPtr) == 0)
            copyFile(tempArgument1,
tempArgument2);
```

Performance Results and Discussion

I will not even try to argue that my implantation is in any way efficient. For each command, if the function had an argument, I created a new array to store that argument without any special characters such as the newline character. When running in Valgrind, it claims that I had 28 mallocs on that test run. Which sounds like a lot to me, and is probably the most I've had in one project. But other than that, it seems to run find and does everything I want it to.

Conclusion

I honestly had fun with this project. I think this is the first time I've had "fun" since 212. I think this may be because I can actually see the result of what my code produced, and it was awesome when everything worked like I wanted it to. There are probably some edge cases I'm not aware of, but overall I'm happy with my project. I learned about system calls, and even though they sound scary, they're no more difficult than anything else.