

Project 1 Midpoint Report

Elena Lucci & Tia Sharpe

Date 9/10/2022

Project Description

- As of right now, the program parses an input string from the user and stores the parsed string for use in part two of the project. In part two, the parsed string will be used to perform commands that will be written into the program.

Achievements

- We used many new or unfamiliar functions that we were not used to using such as; `clear()`, `strtok()`, and `strcpy()`. The `clear()` function allows us to clear the screen when the shell is executed so that the user is not overwhelmed with previous information. The `strtok()` allows us to tokenize the input string based on a certain delimiter. The `strcpy()` function allows us to copy the input string to be manipulated in the `Parse.cpp` class. One of the problems that we encountered was how to parse through the string and specifically find the `'&'` symbol and the end of the string. We had another issue with taking the `'<'` or `'>'` sign off of the token so that we could store the input and output redirect variables without those symbols.
- Includes: `iostream`, `string`, `cstring`, `stdlib.h`, `bits/stdc++.h`, and `algorithm`.

Preliminary Testing

- Most of our testing was mainly in our `Parse.cpp` file. `Param` is basically just a bunch of getters and setters so there wasn't much to test there. A lot of our small tests are still in the code, just commented out because we like to go back and reuse them often.
- We had one test that simply printed out the token twice if it had a `'<'` or `'>'` symbol in front of it.
- We had another test that printed out the tokens as they were being created, we would also print what placement in the `argumentVector` each token was at.
- In an attempt to take off the `'<'` or `'>'` symbol in the token, we created a while loop that continuously added every character of the token after the `'<'` or `'>'` symbols onto a copy. Every time the loop reiterated we printed out the copy and would look at the progression. We are still having issues with this part currently.

Next Steps

- Create a basic shell program that uses the parsed string. Input/output redirection and foreground/background handling will be implemented. We will have to incorporate the `fork()` and `exec()` commands in order to successfully complete this program.