Project 1 Midpoint Report

Jack Lovett

9/12/2023

Project Description

This portion of the project is responsible for prompting the user, parsing user inputs, and exiting when the user requests. The parsed user inputs are saved in a data structure that will be used in part 2 to create and execute commands.

Achievements

I have created the shell prompt, created the Param class, and created some fundamental functions therein. The shell prompt parses a line of user input and correctly directs the data into the corresponding variables. I took an entire line of input from the user and passed this to sstream. From there, I parsed each string as it came out of the string stream’s buffer. I’ve also made use of cstring’s c\_str() function to convert strings to char\*s, and strcpy() to copy the values in those char pointers to the char pointers for inputRedirect and outputRedirect. I’ve also had to use strdup() to allocate new memory and duplicate the user’s inputs. When I was getting the user’s inputs initially I was using a temp char\*, and when I saved that pointer location to argumentVector, every element of argumentVector would point at the last token that the user input.

I also attempted to make a separate parser class in which the only function would parse a given line of input and return a pointer to a new Param object. I had issues returning this pointer, and so in the interest of time I moved the function to myshell.cpp where it functioned as intended.

I haven’t had time (or instruction) on how to make a destructor yet, however we are covering that in my Data Structures and Algorithms class, so I should be able to figure that out shortly.

For some reason, my check for the debug argument is failing. It was working several versions ago, but now when I check if “-Debug” == “-Debug” the check fails. I haven’t changed anything to do with this check and I’m not sure why it stopped working.

Preliminary Testing

Enter command> <a.in >b.out & z y x w v

InputRedirect: [a.in]

OutputRedirect: [b.out]

Background: [1]

ArgumentVector[0]: [z]

ArgumentVector[1]: [y]

ArgumentVector[2]: [x]

ArgumentVector[3]: [w]

ArgumentVector[4]: [v]

Enter command> <a.in 123 423

InputRedirect: [a.in]

ArgumentVector[0]: [123]

ArgumentVector[1]: [423]

Enter command> ls

ArgumentVector[0]: [ls]

For each of these tests, tokens prepended with < were saved to the inputRedirect, tokens with > were assigned to outputRedirect, the & was recognized and the background flag was appropriately set, and the remaining tokens were assigned to ArgumentVector. The program appears to be functioning as intended.

Next Steps

My next steps will be to finish creating accessors and a destructor for the Param class. After that, I will likely attempt to create a new class, or at least separate functions, to handle the command execution. I need to investigate which exec() functions I need to use, and when I should prefer one to the other. I still need to learn how to do input/output from a file, and I have to figure out how to execute the given commands in the background when requested. My biggest problem here is that I don’t know what I don’t know; I’m figuring most of this out as I go along.