

CS 4414 Project 1 Write up
Team Member
Neeraj Gandhi
Wright Kim

Within the infinite while loop of shell.cpp, the program prompts user to enter input and stores it using `getline()`. If this input is longer than 80, the program prints an error message to the console, else, returns the input. Then, shell.cpp compares input with string "error". If the input is not "error", proceeds.

The next step is validating the input using customized function called `validate()`. This functions uses regular expression to check if the characters of the input are valid or not and returns Boolean value. If `validate()` returns true, the program proceeds and check if the input is "exit". If it is, the program terminates itself.

Then the shell.cpp runs `validatePipeIOPlacement()` function. This functions receives user input as an argument and checks if an operator is part of a word, and it prints out an error message if it is.

If the input passed the `validatePipeIOPlacement()`, shell.cpp splits the input using a space as delimiter and pushes into a vector of string. And before calling the `processTokensIteratively()` function which processes the command, it checks whether the commands are valid or not using `validateFileIO()` function.

The function `processTokensIteratively()` receives two arguments; vector of strings, which stores commands in vector as a string. This function then extracts a token and convert it to a C String and push it to the vector of C String. This step is necessary because without converting to C String, commands will not be executed in any exec family functions.

The function `processTokensIteratively()` is now ready to run the process by calling `fork()` system call, which will create a child process. The child process first checks if there is an input or an output redirection within the token. If it doesn't, the process opens the file that contains the commands and sends it to the pipe.

If there is an input or output redirection, the child process should open a `inFile` and `outFile` and map it to the pipe using `dup2()`. However, we were not able to finish this process.