

Programming Assignment 2 Report

Introduction

For this assignment we were tasked to implement our own Unix/Linux shell.

Parsing

Parsing is done in 3 steps

Step 1: `tokenizeString(input,delim)`

Takes the entire user string, and separates it by spaces. Content inside single or double quotes is not split. Results are put in a vector of strings.

Step 2: `splitBySpecials(tokens,specials)`

Sometimes input comes like `ps | head -5`. I can't really work with that, so the tokens are split further based on `|` `<` and `>`. If the splitting symbols are inside quotes, they are not split on. Returns a vector of strings.

Step 3: `removeOccurrencesOf(tokens,'\')`

This step removes single quotes from all the tokens because the `awk` command does not like them.

Piping

It's all done in the `evaluateCommand()` function. I create a `fd[2]` and feed that to `pipe()`. Then I use this single pipe for the remainder of the program. A command to the left of the pipe writes into the pipe write end, and the command on the right of the pipe reads from the pipe read end. Because I split the tokens so much I collect the ones on the left in a new array, and the ones on the right of the pipe in another. Then I execute the left part first, wait for the child to return so I know the contents have been put in the pipe buffer, then I run what's right of the pipe. If there are more pipes, I feed the output of the right side into

pipe write-side again, and it is read from pipe-read side by the next command. If there are no more pipes, the last command writes to standard out.

Redirection

I only implemented redirection in the > direction. You can at the present moment only write to files, and not read. All redirection goes through doRedirect() which calls executeRedirect().

If a simple redirection like `ps > test.txt` is requested, a file is opened and the standard output said to point to the file. Then the redirect function runs `ps`. The redirect function always runs the commands in front of it, and feeds them into the file by setting the standard out to point to the file. If there is a pipe before the redirection, standard in is set to point to the pipe read side, the commands are executed, and the result of that is fed to the file.