COP 4634 PROJECT 1 MIDPOINT REPORT

Title:

COP 4634 Project 1 - Midpoint, Creating Your Own Shell

Description:

Part one of a two part project to implement program myshell. This program will parse an input string into a structure that will be used by part 2 to create a new process. It implements a simplified/basic shell that parses a command entered by the user, input redirection, output redirection and background execution.

Input/Output Redirection Format Fielded:

Program does not accept spaces after the redirect token, it must be in a format of ">input.txt" rather than "> input.txt".

Project Members:

Name: Juan Morales-Vargas

Institution: University of West Florida

Email: jdm161@students.uwf.edu

Name: Brian Francis

Institution: University of West Florida

Email: bf15@students.uwf.edu

FILE OVERVIEW

File List:

myshell.cpp

parse.cpp

parse.hpp

Makefile

README

Launch Instructions:

Within Bash, enter the following command after the prompt, then hit enter:

Make myshell

after that, at the prompt, enter the following command (No Debug):

./myshell

Or instead you may, at the prompt enter the following command (Debug):

./myshell -Debug

The program will enter Debug mode; see Program Behavior, item 5 for further details.

Exit Command:

To exit the program, enter the following command after the prompt, then hit enter:

exit

The program will exit and return the user to Bash.

Program Behavior:

1. Displays a prompt, $$$, on *stdout*.
2. Accept a command as a string from the user (input string will terminate with
   a newline character. The program will exit when the command *exit* is entered
3. Parse the input string into tokens, which are single words delimited by one
   or more spaces, tabs or newline characters.
4. Tokens are stored in the fields of class *Param*.
5. While in Debug mode, entering *printParams()* will print the contents of the
   fields stored in class *Param*.
6. Return back to step 1

Sample Run with Debug mode enabled:

```
./myshell -Debug
$$$ one two three <four >five &
InputRedirect: [four]
OutputRedirect: [five]
Background: [1]
ArgumentCount: [3]
ArgumentVector[0]: [one]
ArgumentVector[1]: [two]
ArgumentVector[2]: [three]
```