Dalton Meny

CS 470

Shell Project

Discussion & Answers to Questions

At the highest level, my program is a while loop that continues until the user enters exit. Within that loop there are various code segments for reading in input, determining if said input calls a special function, and finally takes the input and executes using execvp(). The input is stored as a vector of strings. Then that vector is stored in another vector that holds the history. This makes it easier to access a specific previous command and print the history due to the fact that you can access the information of the vector at any index and that vectors know their size. It also makes it easy to add a new command the history by using push_back().

There were a few problems that I encountered during this project. I foolishly started to write this project in C instead of C++. I forgot how nice C++ strings and vectors were. Even though I had to redo some of the project, it was much easier to write in C++. As for my code, there are a few things that I do not like. At times I feel there are too many if statements, especially nested if statements. Although I don't know if this is avoidable or not. Also, there is probably a better way of converting the vector of strings into a C array, but what I have works. The & functionality was a little difficult, but I got it working in the end. That was definitely the hardest part to understand. On the flip side, I found the execvp() function very easy to understand. I found it interesting how easy it was to create a shell. It did not take much code at all.

Test Plan

!! – “No commands in history”
!1– “No such command in history”
history– “No commands in history”
ls - displays files (single command)
cat prog.cpp – prints the program to the terminal (two commands)
!3– “No such command in history”
!0 – “No such command in history”
!anystring – “No such command in history”
emacs prog.cpp – opens a program (new window)
emacs prog.cpp & - opens new program in background
history– prints 1-4
ls -l – shows files with extra info
cal – prints calendar
date – prints date
ps – shows processes
ls – displays files
more Makefile – prints makefile to output
who – prints out user
ls – displays files
history– prints 3 - 12
!! – executes ls
!12– executes ls (last command)
!2– executes cat prog.cpp
!4 – executes emacs prog.cpp &
___more____Makefile__ (underscore indicates extra spaces) – executes cat prog.cpp
history – prints 4-13 (13 has no extra spaces)
exit – exits program