Clayton Brutus
CS470
Dr. Hwang
Shell Project
2/3/17

Program has a modular design and is written in C++. The program reads command arguments into strings and stores them in a vector, then converts that to an array of C strings for passing to execvp. A vector of vectors of strings is used to past 10 command arguments for history functionality. An integer variable stores the current command number for use when displaying the command number for each command in history, and for executing the Nth command when the "!N" command is used.

Command	Expected Result
<command>	executes command in terminal
<command> &	executes command in background
cd <directory>	changes current working directory to specified directory
pwd	displays path to current working directory
cat <filename>	displays file contents on terminal
ls	lists files and directories in current working directory
ls <directory>	lists files and directories in specified directory
nano <filename>	opens or creates a text file for editing in nano
emacs <filename>	opens or creates a text file for editing in emacs
cp <file1> <file2>	creates a copy of file1 with name file2
mv <filename> <directory>	moves file to specified directory
mv <file1> <file2>	renames file1 to file2

1. What aspect of process manipulation did you find most difficult to understand?
    a. How execvp affects the process it is executed in.
2. What aspect of process manipulation did you find least difficult to understand?
    a. How to use fork to create and use a child process
3. What, if anything, would you change in your current design?
    a. Use more functions to make the main function easier to follow.
    b. Find a better way to give arguments to execvp without converting from strings to c-strings.
4. What, if anything, did you find interesting or surprising about process manipulation that you did not know before doing this project?
    a. I was surprised when learning about how running execvp in a process replaces the process and should not return if there is not an error.