Munachimso Aneke

CS 470

Shell Project 1 (Written Portion)

DISCUSSION

The program is written in C. Majority of the program is in the first while loop of the main. The while loop prints out the user prompt, which is read in as a string. The string is added to an array of string to be stored as a history. Using strtok(), the string is split with the delimiter " " and each substring is added as elements to the array of string args. The command — args— is then checked to see if it is to exit or contains '&'. If the command is to exit, it changes the loop control variable to false, ending the loop. If the command does not contains '&', it turns on the flag to notify the parent process to wait, while the child process runs. The last part of the main section checks if the code is a special command, like history, or it forks a process.

For the history feature, a struct is created to store the commands used. The structs contains an array of strings and counter to store the position of a "cursor." The counter helps to keep track of the number of commands in the string and the position of the last string. A void function called add_history() is created to add new command to the array and increment the counter by 1. A void function, display_history(), takes in a flag and nth position to decide what part of the history to show. The flag is decided based on the command (history, history !!, history ! n). And the nth position determined by the nth position the user requests. If there's no history or nth is more than the number of used commands, a message will be printed out. The history is printed out using a for loop to go back as many times as specified (if valid). The history feature is implemented in the extra_command() function which, depending on the command given, will call the right display_history() condition.

My program does not meet all the requirement. The major part of this program is the creation and execution of a program in a loop. My program runs the loop once before printing out "segmentation fault (core dumped)" and I can seem to spot the error.

TEST PLAN

Command: history

Result: "No history"

Command: ls -l &

Result: list of all the files in the current folder

Command: cal

Result: a calendar and "Child complete"

Command: date

Result: the current date and "Child complete"

Command: history

Result: “4 date”, “3 cal”, “2 ls -l &”, “1 history”

Command: history !!

Result: “history”

Command: history ! 4

Result: “4 history !!”, “3 history”, “2 date”, “1 cal”

Command: history ! 9

Result: “Not up to 9 command in history”

ADDITION QUESTION

1. Execvp() function. My function call seems to not to run any of the commands and the loop runs into a segmentation call, when it runs the 2nd time.
2. Creating the fork process,, and seeting the function call to wait if the child process does not run concurrently.
3. My code for breaking the string into args for execution seems to be a little confusing.Also the way the history is stored.
4. I thought I fully understood shell and using of fork(), exec() and wait() but this project proved otherwise.