Programer: Chau Nguyen

tug37553@temple.edu

This shell program included:

utility.c

utility.h

myshell.c

myshell.h

readme

Command you can use:

cds ,clr, dir, environs, echos, helps, pause, quit

> < and ^(append on file)

Once you are in my shell folder, you can type make and ./test to start the process

User will type the list of the commands: cds ,clr, dir, environs, echos, helps, pause, quit

> < and ^(append on file) to run the program.

To test >: echo Hello > test1

To test < : echo Hi1 < echo hi2

To test >>: echo Add ^ test1

To test | : echo Hi | echo Hello

\_\_\_\_\_\_\_

Utility.c

This utility.c file contain 8 built-in commands that user can use

These functions will be called in the main file

1. cd command

If nothing after cds then bring user to Home directory

Print error if getenv("HOME") is not success

2. clr command

Using this command to print white space and move the pointer on top

printf("\033[2J\033[1;1H");

3. dir command

This dir function will do:

If user just type dir , it call getcwd to get current directory

Then opendir to get info of all the file

The read the name of the file from a built in struct call dirent

If user type dir directory, then it will opendir and get info of all the file

4. environ command

Get environment variable

This function used a built in variable call extern char \*\*environ

5. echo command

If noth arg after echo than do nothing, else just print all the agrs

6. quit command

return exit(1) whenever it was called

7. help command

print read me file

8. pause command

while loop run until user press enter

9. my\_cmds

This array contains the 8 commands that user can type

This is be a triger to call the 8 built commands

10. my\_funcs

Array will be call whenever user input matched with one of the

8 commands in my\_cmds

\_\_\_\_\_\_\_

Myshell.c

This main function will handle execute, redirection, piping

1. main

Main function contain while loop run until the execute return error

This will get get current directory, read user input,

break user input into separate args, call Execute funtion and return

Free malloc

2. getCommandLine

This fuction will read user input everything until EOF ot \n

3. ParseTheCommand

This function will get the whole line and break into pieces

4. detect\_symbol

Find out the | < > or other characters

This will retur

n 0 for pip, 1 for output redirection and 2 for input redirection

5. get\_pos

Get the position of the special characters

6. startprocess

Create process by forking

First it will flag background to be 0

If user enter & at the command, then set bg to true

Fork to create child process

The child will call execute the command

if bg is true then do thing

if bg is false then make parent wait

by calling waitpid(pid , &status , WUNTRACED);

7. piping

fork first time: child duplicate file descriptor to write from stdout

close the pipe and execute the arg1

In the parent, fork again, the child in the parent duplicate fd to read from stdin

close the pipe and execute the arg2

In the parent’s parent, close 2 sides of the pipe

8. execute

Get the special symbol and position that the symbol located

If it's not < > | and if arg match to 8 internal commands

Then call the internal funcs and create the process

Else it < > |

get arg1 [special character] arg1

call 4 cases with piping and redirect function

9. redirection

This will fork and create a child. This child will run one

of 3 cases > < >>