Shell

OS lab 1

name :mina sameh labib
id :75

Contents

Code Organization

the shell consists of 3 main parts:

- **get the instruction**

  Done by having a file to read line by line or user input

- **parse it**

  Splits the coming command to separate the instruction from the parameters and handling special cases like multiple whitespace without a quote, leading or trailing whitespace

- **execute it**

  At execution point the shell creates a new child process then it makes a system call from it to the OS to execute the command with it's parameters , also it handles special cases like : CD, EXIT, log, history and help

Main Functions

after a command is read it will be passed to the following functions

1. void trim(char *str)

   used to delete leading or trailing whitespace

   ex: cd ~ ====>cd ~

2. char * substitute(char *str)

   used to change variables names with their values and also to convert ~ character to the home path

   ex: cd ~ === >cd /home/[user]

   x=5,echo $x===>echo 5

3. int split(char **strArray, char* l)

   used to split the current command to strArray ignoring multiple delimiter

ex: ls -a -l =====> {“ls”, “-a”, “-l”}

then the shell create a child process throw fork(); then it call a function to get the physical path to the command and make a system call from it to the OS to execute the command with execv(); then the result is printed in stdout

other functions are

1. void sig_handler(int signo)

   a function to handle signals sent from children processes to save progress in the log file

2. void getFilePath(char ** name, char * path)

   takes the name of a command and try to find it in any of the PATH locations

3. void printHelp()

prints help message to the user

4. void writeToFile(char * filePath, char * text)

   append text to the end of file specified by filePath used to write log and history values

5. void printFileContent(char * filePath)

   print the contents of a file to the screen used to print log file and history file

How To Compile the Project

1. start terminal
2. change directory to the folder containing the source code and the makefile
3. run make command in terminal
4. you can run the shell by typing

   ./shell [batchFile]