# 

# *Name : Yahia Mohamed Elshahawy ID : 84 CSED 19 Interactive Shell*

# ***1 - Problem Statement:***

A Unix shell is a command-line interpreter that provides a traditional user interface for the Unix

operating system and for Unix-like systems.

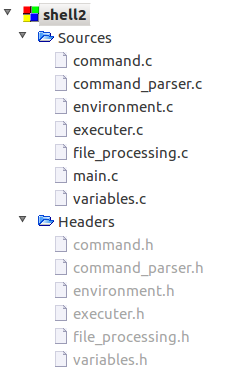
The shell can run in two modes: interactive and batch.

In the shell interactive mode, you start the shell program, which displays a prompt (e.g. Shell>)

and the user of the shell types commands at the prompt.

# ***2 – Code Organization:***

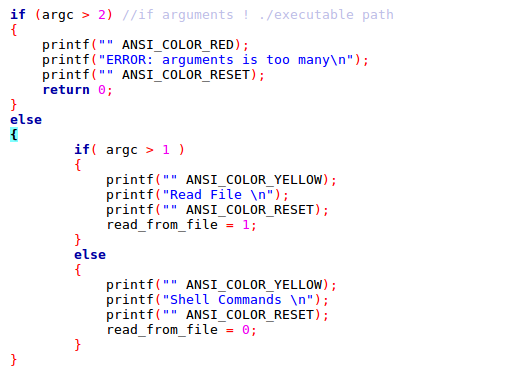
***Project heirarchy:***



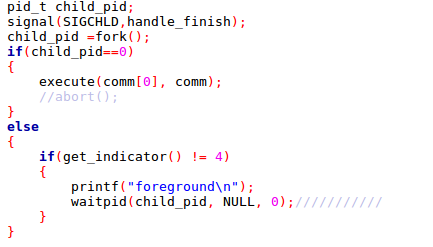
Main Modules:

**1- Main:**

main is the backbone of the project it has main responsabilities :

1- Decides what mode the user has chosen

2- Provide forking of main process to execute commands using the executer



3- Provide some user interface:

by using function: print\_Welcoming() , and some color constants.  
  
MAIN FUNCTIONS:  
1- void handle\_finish() : print to log file when SIGCHLD signal is detected.  
2- void print\_Welcoming() : print to the console some useri nterface words  
3- int main(int argc, char \*argv[]) : it recieves the paramaters entered by the user , choose the mode of Shell, handle exit , call other methods and forking the main process.

**2- Command.o:**

**handle some special commands like cd.  
MAIN FUNCTIONS:  
1- void cd( char\* path ) : take a Path as a parameter , used to change current directory by handling all cd cases.**

**3- command\_parser.o:  
- This module should be responsible for importing all details of the command**

**- Should specify the type of the command "comment, cd, echo, expression - X=5 -, else"**

**- Should specify the arguments of the command**

**- Should specify if the command is background or foreground**

**- Should consider all parsing special cases, example: many spaces in "ls -a"**

**MAIN FUNCTIONS:**

**1- void RemoveTabs(char\* source): remove all tabs from a command.**

**2-int Count\_Occurences(char\* source): count how many times a char is found in the string.**

**3-void handle\_Home\_sign():handle ‘~’ occurences.**

**4-void handle\_dollars(): handle ‘$’ occurences.**

**5-int check\_empty(char\* source): check if an instruction contains only spaces.**

**6-void handle\_var\_set(): handle expressions of “x=value”.  
7-void handle\_bgd(): handle occurences of ‘&’.**

**8-void parse\_command( char\* command ): handle comments , and quotations “” , and tokenize the command.**

**9-int get\_indicator(): get type of command : 2 is normal , 0 is comment , 3 is set variable , 4 is background instruction.**

**10-int get\_command\_length(): get how many tokens located in command.  
11-void get\_clean\_command(char\*\* a): return array of tokens of the command.**

**-----------------------------------------**

**Environment.o:**

**Main Functions:**

**void setup\_environment( void ): This function should be responsible for importing environment variables into your project.**

**Typically, this function should add $PATH, $HOME & any other needed variables into your variables table and $PROJECT which conatains current project directory**

**-----------------------------------------**

**executer.o:**

**Main Functions:**

**int execute(char\* program, char\*\* arg\_list): used to execute commands using execv for normal commands except cd , echo ,history and exit.   
By looping on the path and trying the command in every directory.  
Also use execvp for echo.  
--------------------------------------------**

**file\_processing.o:**

**Main use is to open,get or close (history file, log file and Batch file)  
----------------------------------------------  
variables.o:**

**contains variables table and used to manipulate it  
main functions :**

**void Constructor(): initialize the table by setting basic environment variables.  
char\* lookup\_variable( char\* key ): search for a variable in the table return NULL if not located**

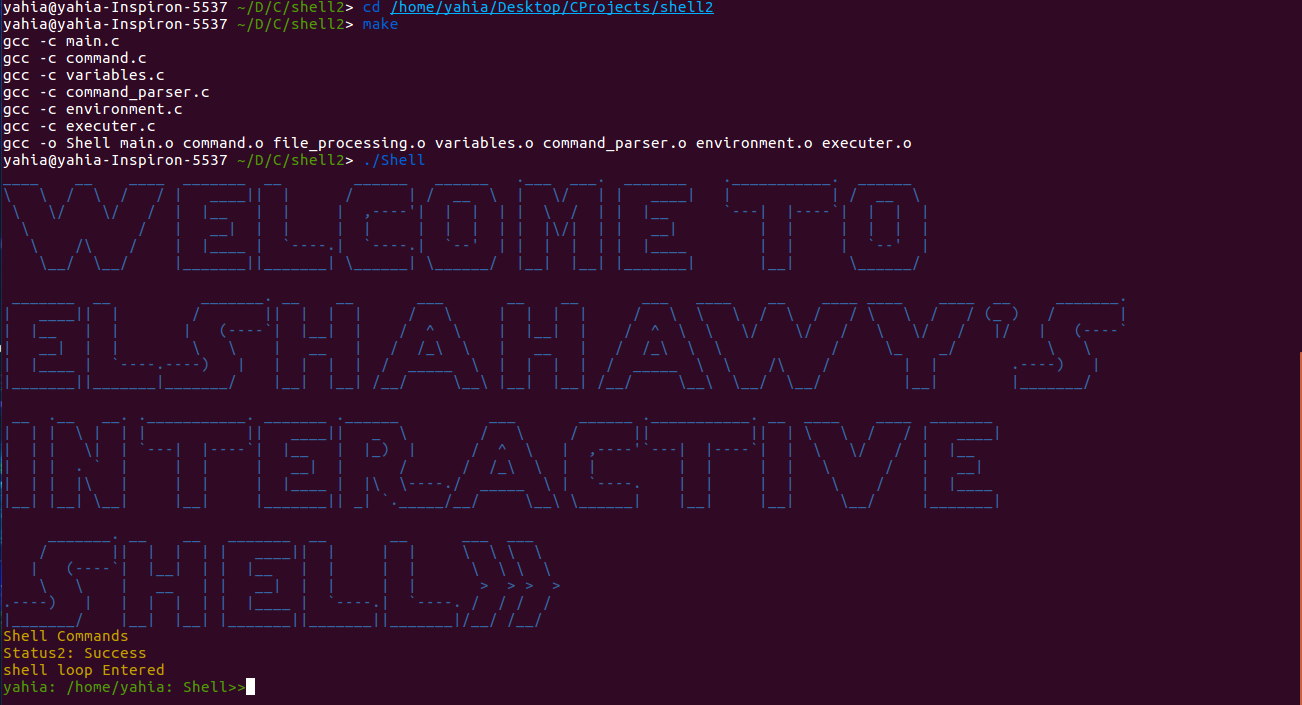
**void set\_variable( char\* key, char\* value ): set value of variable in table**

***3-how to compile and run the code:***

***First: you should expand the terminal window size for good user interface.***

***Second:make sure you are in the same directory of the project***

***third: type make to compile.***

***Fourth:type ./Shell to start in interactive mode or type ./Shell [Path of Batchfile] to start running the batch file then continue with interactive mode***