Name: Ahmed Mohamed Fathallah. ID: 15.

Lab1 Report

Code Organization:

The program sequence goes as following. When the program is run it tries to open the batch file if one was given, if not it runs on interactive mode.

The only difference between 2 modes is the input source and the additional prompt in interactive mode – Shell > -.

Then the input line goes through parsing that divides it into arguments. If the line is empty we go to the next line. If the command was exit we break the while loop and terminate the program. If the command is history we print the history of the user. Other than that there are 2 cases left, either an expression or a command. If it was an expression it's evaluated and stored as a variable.

If it's a command we go through the arguments to replace the variables in form ($x) with their values. Then a child process is created in order to perform the invoked command. A log file is kept indicating the death of each process.

Each input line is kept in history of the user, saved in a file before terminating of the program and retrieved at start.

Each command is either performed in foreground or background. This is determined by the parsing function.

In case of foreground the parent process waits until its child is terminated.

In case of background the parent process continues running along with its child process.

Main Functions :

Parse() :

- Splits the input line into arguments and store them in a string array indicating the number of arguments.
- Splits them on white spaces, taking care of multiple spaces between different arguments.
- A null is saved in the end of the string array in order to be sent to execv() method later.
- Doesn't split on spaces in two cases:
    - Cd command, where the path is kept as it's invoked and only ~ are replaced with /home.
    - Echo method where the argument invoked is a string of form "text", the text inside the quotations is kept unchanged.
- Checks if the last argument is & sign which means that the background flag must be on.

checkExpression():

- Is called to check if the input line is variable declaration or a regular command.
- The input line must have one equal sign.
- The variable on the left hand side must consist of alphabetic characters, decimal digits, or underscore. And it must start with an alphabetic character.
- The value on the right hand side must be a number or a string not containing spaces or a string containing anything but included between quotations.
- If found to be an expression the line is sent to storeVariable() function.
- The helping function goes through the already existing variables and update the value of the given variable if it exists. Other than that a new variable is added to the list.

ReplaceVariables():

- This method is responsible for replacing variables of form $x by their real stored values.
- Is called after parsing the arguments. Goes through the arguments and if one was of form $name, it searches through the existing variables for that name and replace this argument with its value.
- If name was not found the argument is set to null.
- In case of cd command, the argument is later checked to replace each ~ with /home.

CreateProcess():

- Is called after processing the input by parsing it into arguments and replacing variables in it.
- If the command invoked is cd, no process is created because if we change the directory in child process we can't retrieve it to the parent. So in this case the argument is passed to chdir() function inside the parent process.
- Other than that a child process is created and arguments are sent to another function to excute them.
- The parent process waits for the child to terminate if the background flag was off. Otherwise the parent process continues running along with the child.
- If the value returned from the execution function was -1 it indicates an error. So the error is printed using perror(). And it is retrieved from errno.

Execute():

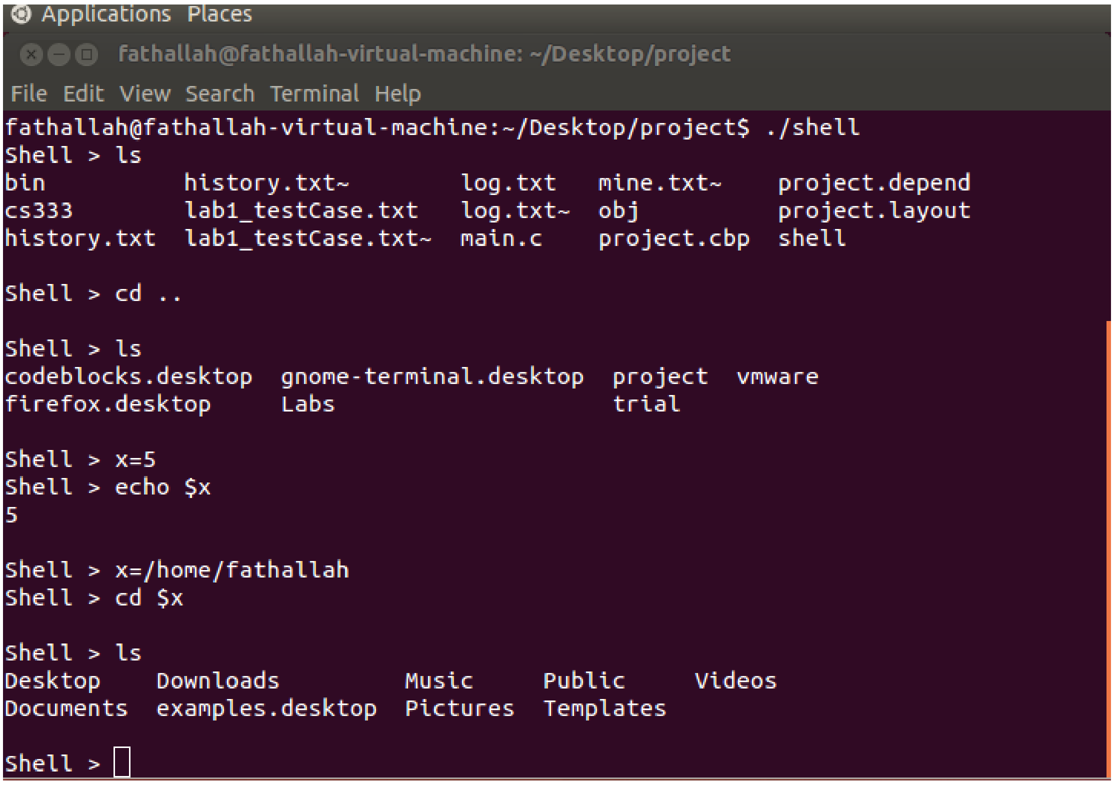
- The arguments are passed inside this function to execv() built in function.
- The first parameter is the path which is found by getenv() function.
- The second parameter is the arguments array prepared before through parsing.
- The function return -1 if execv() function returned -1 indicating an error. Otherwise returns 0.

How to compile and run the code:

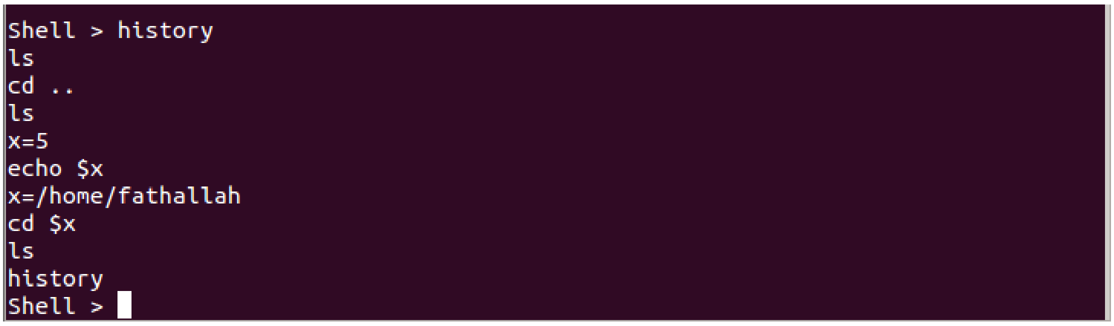
- Through the terminal, change the directory to the sent folder.
- Type make which will run the Makefile.
- An executable file named shell will be created.
- Invoke the file as in : ./shell [batch file name]
- The batch file name is optional.

<u>Sample runs :</u>

Interactive mode

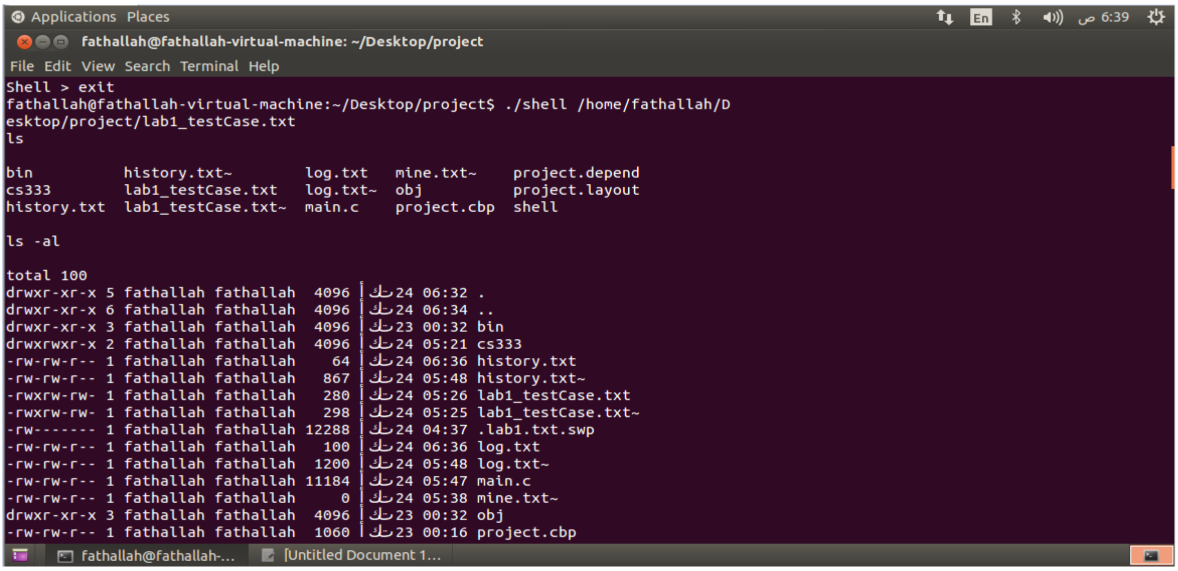

Applications Places
fathallah@fathallah-virtual-machine: ~/Desktop/project
File Edit View Search Terminal Help

```
fathallah@fathallah-virtual-machine:~/Desktop/project$ ./shell
Shell > ls
bin          history.txt~        log.txt   mine.txt~    project.depend
cs333        lab1_testCase.txt   log.txt~  obj          project.layout
history.txt  lab1_testCase.txt~  main.c    project.cbp  shell

Shell > cd ..

Shell > ls
codeblocks.desktop  gnome-terminal.desktop  project  vmware
firefox.desktop     Labs                    trial

Shell > x=5
Shell > echo $x
5

Shell > x=/home/fathallah
Shell > cd $x

Shell > ls
Desktop    Downloads         Music     Public     Videos
Documents  examples.desktop  Pictures  Templates

Shell > 
```



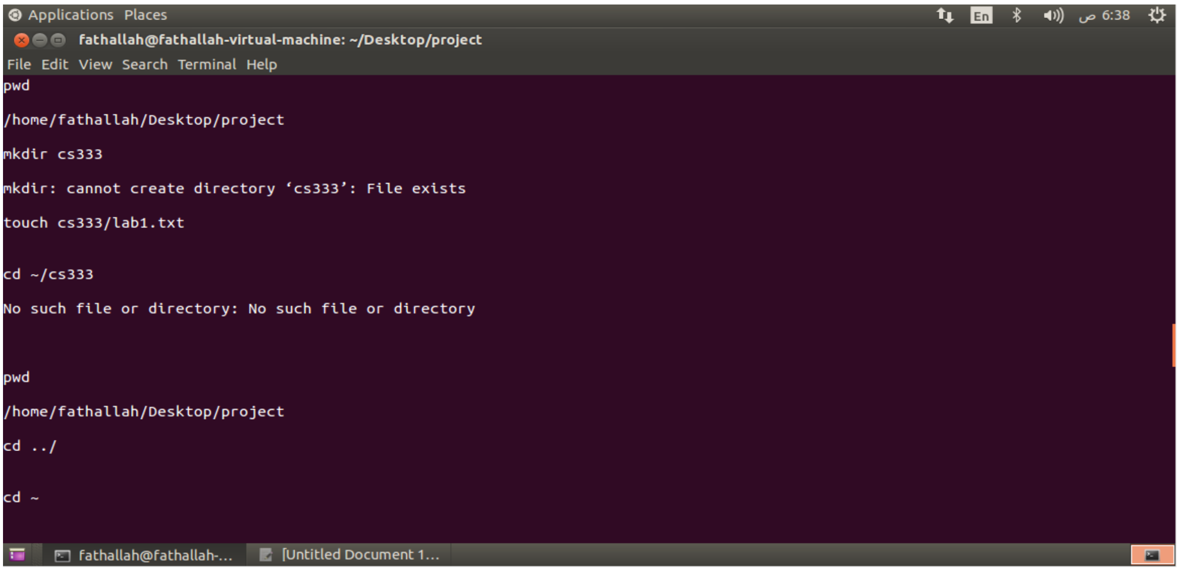
```
Shell > history
ls
cd ..
ls
x=5
echo $x
x=/home/fathallah
cd $x
ls
history
Shell > 
```

Batch mode



Applications Places
6:39 ص
fathallah@fathallah-virtual-machine: ~/Desktop/project
File Edit View Search Terminal Help
fathallah@fathallah-...
[Untitled Document 1...

```
Shell > exit
fathallah@fathallah-virtual-machine:~/Desktop/project$ ./shell /home/fathallah/D
esktop/project/lab1_testCase.txt
ls

bin          history.txt~        log.txt   mine.txt~     project.depend
cs333        lab1_testCase.txt   log.txt~  obj           project.layout
history.txt  lab1_testCase.txt~  main.c    project.cbp   shell

ls -al

total 100
drwxr-xr-x 5 fathallah fathallah  4096 اكت24 06:32 .
drwxr-xr-x 6 fathallah fathallah  4096 اكت24 06:34 ..
drwxr-xr-x 3 fathallah fathallah  4096 اكت23 00:32 bin
drwxrwxr-x 2 fathallah fathallah  4096 اكت24 05:21 cs333
-rw-rw-r-- 1 fathallah fathallah    64 اكت24 06:36 history.txt
-rw-rw-r-- 1 fathallah fathallah   867 اكت24 05:48 history.txt~
-rwxrw-rw- 1 fathallah fathallah   280 اكت24 05:26 lab1_testCase.txt
-rwxrw-rw- 1 fathallah fathallah   298 اكت24 05:25 lab1_testCase.txt~
-rw------- 1 fathallah fathallah 12288 اكت24 04:37 .lab1.txt.swp
-rw-rw-r-- 1 fathallah fathallah   100 اكت24 06:36 log.txt
-rw-rw-r-- 1 fathallah fathallah  1200 اكت24 05:48 log.txt~
-rw-rw-r-- 1 fathallah fathallah 11184 اكت24 05:47 main.c
-rw-rw-r-- 1 fathallah fathallah     0 اكت24 05:38 mine.txt~
drwxr-xr-x 3 fathallah fathallah  4096 اكت23 00:32 obj
-rw-rw-r-- 1 fathallah fathallah  1060 اكت23 00:16 project.cbp
```



Applications Places
6:38 ص
fathallah@fathallah-virtual-machine: ~/Desktop/project
File Edit View Search Terminal Help
fathallah@fathallah-...
[Untitled Document 1...

```
pwd

/home/fathallah/Desktop/project

mkdir cs333

mkdir: cannot create directory ‘cs333’: File exists

touch cs333/lab1.txt


cd ~/cs333

No such file or directory: No such file or directory



pwd

/home/fathallah/Desktop/project

cd ../


cd ~
```