

CSE231 - Operating Systems

Assignment 1

Basic Linux/Unix Shell

Name – Vibhu Jain

Roll No – 2020151

Section – A

Branch – CSE

To the run the program, run the following make command in terminal:

Step 1 -> make external

Step 2 -> make

Shell Interface:

```
Simple Shell : Made by -Vibhu Jain
```

```
USER is: @vibhu02jain  
>|
```

Here after > we have to write our command that is being implemented.

Command Implemented:

Internal Commands :

1. **Echo** – The echo command takes in arguments in the echo [options] [args] format and then prints out the args in the terminal.

Options:

- -n : For not going to a new line on executing echo.
- -e : enables the interpretation of backslash escapes
 - **backslash escapes implemented are:
 - \n : inserts new line
 - \c : supresses trailing new line

```
USER is: @vibhu02jain
>echo hello
hello
>echo -n hello
hello >echo hello
hello
>echo -e hello \nGuys \chi
hello
Guys
>
```

Case handling :

- When No arguments is given after echo.
- When wrong flag is given . [Gives invalid command:success]

```
USER is: @vibhu02jain
>echo
Please also give arguments...!!
>echo -q hello
invalid command: Success
>echo --n hello
invalid command: Success
>
```

Testcases:

- echo hello world
- echo -n hello
- echo -e hello \nGuys \chi

2. **cd** : The cd command is used to change directories. It follows the cd [options] [args] format, where it changes the shell's current directory to the one given in args, if it is accessible.

Options:

- -L : Default state of cd
- -- help : displays the help page
- cd .. : Used to go to the previous Directory ...!!
- cd ~ : Used to go to the home directory ...!! [Not implemented properly]

Default -L is used.

Case handling :

- If no directory mentioned, tells that arguments no given .
- If wrong directory name given, remains in the current directory only.

```
USER is: @vibhu02jain
>cd
Please give the arguments...!!
>cd hello
dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1/simple_shell
>cd ..
dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1
>cd simple_shell
dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1/simple_shell
>cd -L ..
dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1
>cd ~
/home/vibhu02jain
>
```

Testcases :

- cd ...
- cd ~
- cd -L (dir name)
- cd -help
- cd simple_shell

3. **pwd** : The pwd command is used to print out the current directory in which the user is in. It follows the pwd [options] format.

Options:

- -L : Default state of pwd .
- -- help : display the help page

Case Handling:

- If wrong flag is used, it gives Command not found .

```
USER is: @vibhu02jain
>pwd
Current working dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1/simple_shell
>pwd -L
Current working dir: /mnt/c/Users/vibhu/OneDrive/Desktop/iiitdelhi/OperatingSystems/Assignments/Assignment_1/simple_shell
>pwd -t
Command not found
>pwd --help
pwd: pwd [-L][--help]
Print the name of the current working directory.
Options:-L print the value of $PWD if it names the current working directory
By default, 'pwd' behaves as if '-L' were specified.
--help :: Gives the information about the PWD..!!
```

Tescases:

- pwd
- pwd -help
- pwd -L

4. **History** : The history command shows the history of commands executed in the terminal and maintains a permanent storage of these commands. It follows the history [options] format.

Options:

- -c : clears the history
- -s : append the argument to the history list as a single entry

Case Handling:

- If wrong flag is used , it gives the invalid command after it .
- If no arguments given in history -s , it will do nothing . Means it simply needs arguments if we want to append something in history.

```
>history
mys mydate mydate&t history
>history -c
>history
history
>history -s hello
>history
history hello history
```

```
>history -e
invalid command: Success
>history -s
>
```

Testcases:

- history
- history -c
- history -s hello world

External Commands :

I have implemented External Commands using two ways :

1. Using Fork () / execvp()
2. Using Pthread / system()

So to differ between fork() and pthread , I have used “&t” after the command to run that command using pthread. else every thing remains same i.e . Options , flags, erros etc.

So for running “mys”

- mys – This command uses fork() .
- mys&t – This command uses thread.

Commands :

1. **mys** : diplays all the directories and the files in the given directory .

Options :

- -a : show all the folders and files(including hidden folders represented by '.' and '..')
- -s : sorts the contents according to their extension

Case Handling:

- When wrong flag/entries is given , it gives No file or directory found .
- Handles forking error if forking failes. Same for the case of pthread.

Testcases:

- myls
- myls -a
- myls -s

2. **mycat** : The mycat command is used to display the contents of the selected files . It follows the mycat [options] [args] format.

Options:

- -n : display all the output with numbered lines
- -E : adds '\$'to end of every line

Error Handled:

- When no arguments is given , it throws to please give the arguments. This error case is handled.
- When wrong flag is used, it gives invalid command.

Testcases:

- mycat history.txt
- mycat -E history.txt
- mycat -n history.txt

3. **mydate** : The mydate command is used to display the current date and time in accordance with the system's set date. It follows the mydate [options] format.

Options:

- -u : display UTC/GMT time
- -I : display ISO time

Error Handled:

- Invalid flags/entries are checked for and display invalid command if occurrence happens.
- If forking fails it gives Forking Failed error if this happens.

Testcases:

- mydate
- mydate -u
- mydate -I

4. **myrm:** The myrm command is used to remove a particular file from the current directory we are in. It follows the myrm [options] [arg] format.

Options:

- -v : verbose, tells the user which file was deleted
- -i : interactive deletion, asks the user before deleting

Error Handled:

- Invalid flags/entries are checked for and display invalid syntax command if occurrence happens.
- Checks if atleast 1 argument is given .
- Checks if the argument given is a file and not a directory.

Testcases :

- myrm file1
- myrm -v file1
- myrm -i file1

5. **mymkdir:** The mymkdir command is used to create a directory with a given name in the directory we presently are in. The format for mkdir is mymkdir [options] [args] .

Options:

- -v : verbose, tells the user which folder was created .
- --help : gives the help page

Error Handled:

- Checks if atleast 1 argument is given .
- Checks if the directory doesn't exists previously.
- More than 1 folder can't be created together. One at a time .

Testcases :

- mymkdir folder
- mymkdir -v folder
- mymkdir --help

Simple Testcase for whole:

- myls
- history
- mymkdir&t -v vibhu
- cd -L vibhu
- myls
- mydate -I
- cd ..
- pwd
- cd ..
- myls -s
- history -c
- history
- myrm filename.txt
- mydate -u
- cd simple_shell
- mycat history.txt

Thank You