

Purpose:

The project is made to imitate the bash terminal of a UNIX based system like Ubuntu. It comes with its own restrictions and limited flags per command.

This CLI provides the user with 10 explicit commands of which 5 are internal namely :

- 'cd',
- 'echo',
- 'history',
- 'pwd' and
- 'exit'

And 5 external commands namely :

- 'ls',
- 'cat',
- 'date',
- 'rm' and
- 'mkdir'

Each comes with its own set of 2 flags namely :

Commands	Flags
cd	-P and -L
echo	-n and --help (-E is default)
history	-w and -c
pwd	-P and -L
exit	-- no flags there --
ls	-a and -r
cat	-E and -n
date	-u and -R
rm	-v and -d
mkdir	- v and -P

Robustness of errors

- You can input numerous directories to be removed in one go in the rm command but it removes only the files if “-d” is not included in the input command.
- Similarly multiple directories can be made with mkdir when space separated directories are provided in the command input.
- ‘mkdir’ can make directories when a path is specified from the root if the machine provided that directory is not restricted with administrative privileges.
- For every command, if the number of parameters input are not sufficient, then an error message pops up.
- For every command, if a flag is input not include in the above table, then also an error message pops up.
- Space as a delimiter is provided for the user to input multiple directories in rm and mkdir commands as mentioned above.

Assumptions made

- First and foremost assumption is the format of the input command
 - <command> -<flag> <directory> <directory>Any other format might give runtime error or wrong outputs from the one expected.
- No directory name with space can be input. If done so, the programs will not perform the preferred operation in an efficient way.
- Do not provide any directory name with double quotes.
- In mkdir, the nested directories can't be made if the path is specified from the root. But if a single directory is to be made with path specified from the root, that works as per the expectations.
- Nested flag are not implemented as given in the Assignment description.

Working of the commands:

1. cd

Description:

Used to change your working directory.

cd or cd -P or cd -L

Will change to root directory

cd -P <directory path>

Will change to physical path of the directory input

cd -L <directory path>

Will change to logical path of the directory input

cd ..

Will change to parent directory.

cd .

Remains on the current directory

2. echo

Description :

Used print whatever you type on the screen

echo

- Will print everything you type on the terminal after you press the “return key”, and end with a newline.

echo -n <input>

- Will print the <input> but not a newline at the end

echo --help

- Will print all details related to the echo that might come in handy.

3. **history**

Description

Used to present all the commands input by the user so far.

history

- Shows all the commands input by the user so far numbered at each line.

history -w <filename>

- Copies the history to the file name specified by the user.

history -c

- Clears the history

4. **pwd**

Description

Prints the current working directory.

pwd -P

- Print the physical path of the current working directory.

pwd -L <directory>

- Prints the logical path of the current working directory.

pwd <directory>

- Prints the path to the current working directory.

5. **exit**

Description

Will exit the terminal

6. **ls**

Description

Prints the directories in the present directory.

ls

- Prints all the files and folders excluding the hidden directories.

ls -a

- Prints even the hidden folders in the current working directory.

ls -r

- Prints the non-hidden files and folders in reverse lexicographic order.

7. **cat**

Description

Prints the data inside the file provided as parameter.

cat

- Prints the input the user gives to the cli on a newline

cat <filename>

- Prints the contents inside the file input by the user.

`cat -E <filename>`

- Prints the '\$' at the end of each line it prints on the screen after reading the data from the file input by the user.

`cat -n <filename>`

- Prints the line number for each line output by the cli.

Similar working is shown when no filename is provided rather just a flag.

8. **Date**

Description

Prints the date in different time zones as per the flags input.

`date`

- Prints time in IST

`date -u`

- Prints time in UTC

`date -R`

- Prints time in IST showing the difference with UTC

9. **rm**

Description

Removes the file/directory provided to the cli.

`rm <filename>`

- Removes the file input by the user

`rm -v <filename>`

- Removes the filename and prints that the file has been deleted if success.

`rm -d <directory>`

- Removes the directory input to the cli

10. **mkdir**

Description

Used make a directory

`mkdir <directory/path>`

- Will make a directory for you .

`mkdir -P <directory path>`

- Will make nested directories if not present

`mkdir -v <directory>`

- Will make the directory and print that the directory has been made.

Some of the test cases are shown below in screen shots:

```
Activities Terminal
tushar@tushar-HP-Pavilion: ~$ cd /home/tushar/c/OS/Assignment1/ques2
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$ cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ cd ques2/c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2/c$ ls
No such file or directory
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2/c$ cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$ ls
asdasdecho.txt
date.c
rm
date
rm.c
ques2.c
Assignment1_05_2019393_Q2.pdf
mkdir
extt.txt
ls.c
ls
Makefile
cat
history.txt
ques2
mkdir.c
cat.c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$ cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ cd c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/c$ pwd
/home/tushar/c/OS/Assignment1/c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/c$ pwd -P
/home/tushar/c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/c$ pwd -L
/home/tushar/c/OS/Assignment1/ques2
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/c$ cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ cd -P c
TusharMohan_2019393:/home/tushar/c$ ls -la
total 12
drwxr-xr-x 3 tushar tushar 4096 Nov 14 12:35 .
drwxr-xr-x 3 tushar tushar 4096 Nov 14 12:35 ..
-rw-r--r-- 1 tushar tushar  120 Nov 14 12:35 a.out
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 a.s
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 hello.c
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 main1.c
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 a.c
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 Makefile
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 history.txt
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 a
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 a.txt
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 function.c
-rw-r--r-- 1 tushar tushar   21 Nov 14 12:35 OS
TusharMohan_2019393:/home/tushar/c$
```

```
Activities Terminal
tushar@tushar-HP-Pavilion: ~$ cd /home/tushar/c/OS/Assignment1/ques2
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$ history
1 history -d 2
2 cd
3 ls -l
4 asdas
5 history
6 history
7 history
8 as
9
10 asd
11
12 qwe
13
14 history
15
16 qwe
17 qwe
18 fdaf
19 history
20 rm test
21 rm test
22 rm -v cat
23 rm -d asd
24 mkdir tushar
25 mkdir tushar
26 mkdir tushar
27 mkdir tushar
28 mkdir tushar
29 mkdir tushar
30 mkdir tushar
31 mkdir rahsut/tushar/pop
32 mkdir mkdir -v rahsut
33 mkdir -v devu
34 mkdir -v devu
35 mkdir -p qwe/noin/asda
36 mkdir -p oublo/sdfs/werwe
37 mkdir -d asda/fgf/ewe
38 mkdir qwe
39 mkdir qwe/wer/ert
40 mkdir -p qwe/wer/ert
41 mkdir -p /qwe/wer/ert
42 mkdir qwert/qwert/qwert
43 mkdir -p qwert/qwer/qwe
44 mkdir -p asd/asd/as/
45 mkdir -p ghf/dfg/dfg
46 mkdir -p ert/er/er
47 mkdir -p /home/tushar/c/OS/Assignment2/qweev/sdfs/sdfsdf
48 mkdir qweq
49 mkdir qweo
50 mkdir werwer
51 mkdir -p theor/dfgds/sdfsdf
52 mkdir -p /home/tushar/c/OS/Assignment2/qweq/qweq/qweq
53 mkdir -p qwrq/sdfs/sdfs
54 mkdir -v wqeqwe
55 dfds/sdfs/mkdir -p qwesdas/sdfs/sdfs
56 mkdir -p qweq/qweq/qweq
```

```
Activities Terminal
tushar@tushar-HP-Pavilion: ~
File Edit View Search Terminal Help
}
return 0;
}tusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$cat -E rm.c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <dirent.h>
#include <fcntl.h>
$
int main(int x, char** params){
if(params[1]==NULL){
printf("Not enough params\n");
return 0;
}
if(params[2]==NULL){
FILE *fptr= fopen(params[1], "r");
struct stat buf;
stat(params[1], &buf);
if(!fptr || !S_ISREG(buf.st_mode)){
printf("File not found\n");
}
else if(S_ISREG(buf.st_mode)){
remove(params[1]);
// printf("Removed\n");
}
else{
printf("File not found\n");
}
}
else if(params[1][0]!='-'){
int i=1;
while(params[i]!=NULL){
FILE *fptr= fopen(params[i], "r");
struct stat buf;
stat(params[i], &buf);
if(!fptr || !S_ISREG(buf.st_mode)){
printf("File not found\n");
}
else if(S_ISREG(buf.st_mode)){
remove(params[i]);
// printf("Removed\n");
}
else {
printf("File not found\n");
}
}
i++;
}
else if(strcmp(params[1], "-v")){
int i=2;
while(params[i]!=NULL){

```

```
Activities Terminal
tushar@tushar-HP-Pavilion: ~
File Edit View Search Terminal Help
}
return 0;
$
tusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$cat -n rm.c
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <sys/stat.h>
4 #include <unistd.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <errno.h>
8 #include <dirent.h>
9 #include <fcntl.h>
10
11 int main(int x, char** params){
12 if(params[1]==NULL){
13 printf("Not enough params\n");
14 return 0;
15 }
16 if(params[2]==NULL){
17 FILE *fptr= fopen(params[1], "r");
18 struct stat buf;
19 stat(params[1], &buf);
20 if(!fptr || !S_ISREG(buf.st_mode)){
21 printf("File not found\n");
22 }
23 else if(S_ISREG(buf.st_mode)){
24 remove(params[1]);
25 // printf("Removed\n");
26 }
27 else{
28 printf("File not found\n");
29 }
30 }
31 else if(params[1][0]!='-'){
32 int i=1;
33 while(params[i]!=NULL){
34 FILE *fptr= fopen(params[i], "r");
35 struct stat buf;
36 stat(params[i], &buf);
37 if(!fptr || !S_ISREG(buf.st_mode)){
38 printf("File not found\n");
39 }
40 else if(S_ISREG(buf.st_mode)){
41 remove(params[i]);
42 // printf("Removed\n");
43 }
44 else {
45 printf("File not found\n");
46 }
47 i++;
48 }
49 }
50 }
51 else if(strcmp(params[1], "-v")){

```

```
Activities Terminal
tushar@tushar-HP-Pavil Wed 23:46 -/c/OS/Assignment1/ques2

File Edit View Search Terminal Help
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$date
Wed Sep 30 23:45:17 IST 2020
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$date -u
Wed Sep 30 18:15:19 UTC 2020
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$date -R
Wed, 30 Sep 2020 23:45:22 +0530
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$cd A1
No such file or directory
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$mkdir A1
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ls
c
A1
rm
a.out
date
output.txt
test.c
temp.c
ques1
ls
history.txt
cd.c
ques2
cd
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$cd /home/tushar/c/OS/Assignment1/A2
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ls
c
A2
rm
a.out
date
output.txt
test.c
temp.c
ques1
ls
history.txt
cd.c
ques2
cd
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$
```

```
Activities Terminal
tushar@tushar-HP-Pavil Wed 23:49 -/c/OS/Assignment1/ques2

File Edit View Search Terminal Help
TusharMohan_2019393:/home/tushar/c/OS/Assignment1/ques2$cd ..
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ls
c
A2
rm
a.out
date
output.txt
test.c
temp.c
ques1
ls
history.txt
cd.c
ques2
cd
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$rm -d A2
A2
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ls
c
rm
a.out
date
output.txt
test.c
temp.c
ques1
ls
history.txt
cd.c
ques2
cd
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$rm cd.c
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$ls
c
rm
a.out
date
output.txt
test.c
temp.c
ques1
ls
history.txt
ques2
cd
TusharMohan_2019393:/home/tushar/c/OS/Assignment1$
```