DOCUMENTATION

- Type "**make all**" to build all the files and then type "**./a.out**" to start the shell.
- Type **"exit"** to exit the shell.
- **Change the path of the ls,mkdir,rm,date,cat files given in the exec() and system() command according to the absolute path on your system**.

Commands handled:-

1) cd:

cd [dirname} to go to the directory present in the folder.
cd .. to go back to the parent directory.
cd to go to the home directory.

Errors handled:-
If the directory is not present,then an error message is printed.
If a wrong command is used,then an error message is printed

2) pwd:

pwd to print the current working directory.
pwd -L to print the current working directory variables including symbolic links.
pwd -P to print the path to the current working directory

Errors handled:-
If an invalid options (like pwd -c) is passed then an error message is printed.

3) echo:

echo [arg] to print the argument passed after echo command.
echo -n [arg] to print the argument passed after removing any new line character('\n')
echo --help to print the manpage of echo command.

4) ls:

ls to print the files and directories in the current directory.
ls -a to print all the files and directories (including hidden files) in the current directory.
ls -r to print the files and directories in the current directory in reverse order.

Errors handled:-
If no directory or files are present in the current directory,then an error message is printed.
If a wrong command line option is used,then an error message is printed.

5) mkdir:

mkdir [dirname] to create a a directory if not already present.
mkdir -v [dirname] to create a directory if not already present and print a message if the directory has been created succesfully.

mkdir -m [mode] [dirname] to create a directory if not already present and set its mode (or permission) as mode.

Errors handled:-
If the directory is already present,then an error message is printed.
If a wrong command line option is used,then an error message is printed.

Assumptions made:-
Only one directory name will provided at a time.
Commands like "mkdir [dir1] [dir2] [dir3]" are not handled.
Name of the files should not be greater than 7 characters

6) rm:

rm [filename] to remove the given file if present.
rm -i [filename] to remove the given file after confirming once from the user.
rm -f [filename] to remove the given file without confirming/warining to the user.

Errors handled:-
If the file is not present,then an error message is printed.
If a wrong command line option is used,then an error message is printed.

Assumptions made:-
Only one file name will provided at a time.
Commands like "rm [file1] [file2] [file3]" are not handled.
Name of the files should not be greater than 7 characters

7) date:

date to print the current date and time.
date -u to print the date and time according to GMT timezone.
date -r [filename] to print the last modified time of the given file.

Errors handled:-
If the file is not present (in date -r [filename] command),then an error message is printed.
If a wrong command line option is used,then an error message is printed.
Name of the files should not be greater than 7 characters

8) cat:

cat [filename]   to read and print the contents of the given file if present.
cat -n [filename]   to read and print the contents of the given file if present along with the line number.
cat -e [filename]   to read and print the contents of the given file if present along with a '$' character at the end of each line.

Errors handled:-
If the file is not present,then an error message is printed.
If a wrong command line option is used,then an error message is printed.
I have provided "abc.txt" to test the cat command.

NOTE:- While calling execv(path,arg), path is set according to my own location of the file in the system. Like to call the program of "mkdir" I have called "/Users/vinayakarora/Desktop/OS_SHELL_ASSIGNMENT/mkdir". So while running it in another system, change the paths according to the location of the files in your system. I have put comments next to the lines in the code where changes will be required during execution.

TEST CASE

First change all the paths according to your own location of the files.

```
ls
ls -a
ls -r
cd ..
cd [foldername]
pwd
pwd -L
pwd -P
echo hello
echo -n hello
echo --help
```

```
mkdir try
mkdir -v try2
mkdir -m 777 try3
cat abc.txt
cat -n abc.txt
cat -e abc.txt
date
date -u
date -r abc.txt
rm -i  abc.txt
```