

ASSIGNMENT 0

In this assignment, I implemented a shell with some of the commands functioning, both internal as well as external commands.

Internal Commands :

Internal commands are handled directly by the shell itself. List of Internal commands implemented are as follows :

1. Cd :

The cd command is used to change directories.

Options :

- This command can handle the following cases :
 - cd ~ : changes directory to user directory
 - cd / : changes directory to root
 - cd .. : changes directory to previous directory in hierarchy
 - cd . : changes directory to current directory itself
 - cd : changes directory to user directory
 - cd arg : changes directory to subdirectory “arg”, incase if existing.

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - In case if the directory or file passed as argument does not exist, Shell prints the error : “No such file or Directory.”
 - In case if the argument is valid and existing, but is not a directory, thereby making cd command invalid for it, Shell prints the error : “Not a directory.”

Test Cases :

- Following test cases can be considered :
 - `cd ~`
 - `cd /`
 - `cd`
 - `cd .`
 - `cd ..`
 - `cd arg` - arg is a valid existing directory
 - `cd arg1/arg2` - arg1 and arg2 are valid existing directories
 - `cd arg1` - arg1 is not an existing subdirectory
 - `cd arg2` - arg2 is not a directory, although existing

2. Echo :

This command is used to display a line of text on the standard output.

Options :

- This command can handle the following cases :
 - `Echo -n` : This option does not output the trailing new line.
 - `Echo -E` : This option disables the interpretation of backslash escapes.

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - Whether the text is enclosed within double quotes or not, the text is displayed without them on the terminal.
 - In case of an odd number of double quotes, the prompt comes up to take more input unless all opening quotes have a pair of their closing quotes.

Test Cases :

- Following are the test cases that can be considered :
 - `Echo abc`
 - `Echo -n abc`
 - `Echo "abc"`

- Echo -E ab\c
- Echo "ab
>cd
>bjkkl"

3. Pwd :

Pwd stands for Print work directory. It prints the absolute path of the current working directory.

4. History :

This command prints the history of all the commands entered in the Shell.

Options :

- This command can handle the following cases :
 - History : simply displays the past record of commands.
 - History -c : Clears the history log file for that session only.
 - History -a : Appends the commands from the history list to the history file.

Test Cases :

- Following are the test cases that can be considered :
 - History
 - History -c
 - History -a
 - Exit followed by History

5. Exit :

This command simply exits the shell program which was in execution.

External Commands :

1. Date :

Prints the current date and time of the system.

Options :

- This command can handle the following cases :
 - date : IST Format : Prints the current date and time according to Indian Standard Time
 - date -u : UTC Format : Prints the date and time according to GMT i.e Universal time Coordinated format
 - date -R : RFC 2822 Format :
day/date/time/year/hours/minutes/seconds

Test Cases :

- Following are the test cases that can be considered :
 - date
 - date -u
 - date -R

2. Mkdir :

This command is used to make directories.

Options :

- This command can handle the following cases :
 - mkdir -p : Shows no error if existing, else makes parent directories as and when required.
 - mkdir -v : prints a message for each created directory.

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - In case if the directory already exists, Shell prints the error message : "Directory already exists."
 - In case of mkdir -v -p : For all those directories which are already present, it does not show any error. Also it shows the verbose message for only those subdirectories which are created in the process.

Test Cases :

- Following test cases can be considered :

- mkdir a : where a is already present in the current directory.
- mkdir a : where a is not already existing
- mkdir -v a : where a is not already existing
- mkdir -p a/b/c : where a,b,c may or may not exist
- Mkdir -p -v a/b/c: where a,b,c may or may not exist

3. Ls :

This command lists all the contents present in the current directory.

Options :

- This command can handle the following cases :
 - ls -a : does not ignore entries starting with “.”
 - ls -m : fill width with a comma separated list of entries
 - ls : displays all the files except those starting with “.”

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - For a directory with only hidden files (empty directory) :
ls will give “No contents in directory”, while ls -a wont.

Test Cases :

- Following test cases can be considered :
 - ls
 - ls -a
 - ls -m
 - ls -a -m

4. Cat :

Concatenate files and print on standard output.

Options :

- This command can handle the following cases :
 - Cat : simply print the concatenated output of existing files on standard output.

- Cat -n : Numbers all the output lines.
- Cat -b : Numbers only non-empty output line, overrides the -n options.

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - In case if any of the files to be concatenated does not exist, Shell prints the message : “ No such directory or file exists.”

Test Cases :

- Following test cases can be considered :
 - Cat a b : where a is a file that does not exist whereas b exists and is a file.
 - Cat a b: where both a and b are files which exist.
 - Cat a b : where a,b may be a file or a directory
 - Cat a
 - Cat -n a
 - Cat -b a

5. Rm :

This command helps to remove files or directories.

Options :

- This command can handle the following cases :
 - rm -i : prompts before every removal in order to confirm removal.
 - rm -d : removes empty directories.

Corner Cases :

- Following are the corner cases handled, keeping in mind the errors that can occur :
 - rm a , where a is a directory will print error : ” Not a directory ”.
 - rm -i k, where k does not exist is the current directory, prints the error “Not a file or directory”.

Test Cases :

- Following test cases can be considered :
 - Rm a : where a can be an existing or not existing file
 - Rm a : where a can be an existing or not existing directory.
 - Rm -d a : where a is a file or a directory.
 - Rm -i a
 - Rm -i -d a : where a is a file or a directory