

The following commands have been handled in this project.

Internal : cd, echo, pwd, exit

External: ls, date, cat, mkdir, rm

The same applications have been carried out with child processes created through fork as well as through threads. For every command, two variations have been incorporated into the shell along with the default usage of that command. The features of each command and their required syntaxes are listed below.

dir\_name : The name of the relevant directory.

text : Sample text.

fname.txt : The name of a text file present inside the relevant directory.

mode : The mode in which the user wants to make the directory.

## Internally Handled Commands

### Cd

cd	Exits the current working directory and redirects to home.
cd ..	Exits the current working directory and goes to the directory directly outside.
cd 'dir_name'	Enters into the directory as specified.

#### Corner Cases:

- 1) If the user wants to operate any of these commands after entering a particular directory through cd, they can still access the original files present in the original directory.
- 2) It will return a statement indicating if a directory does not exist in case a user tries to access one.

### Echo

echo text	Returns the string entered by the user.
echo -n text	Returns the string entered by the user without a trailing new line character.
echo --help	Explains the features of the 'echo' command.

echo -E text also carries out the default feature.

#### Corner Cases:

- 1) If the user does not enter text, the terminal will display a message indicating them to retry.
- 2) If the user tries to enter blank space, the terminal will prompt them to try again instead of returning a segmentation fault.

### Pwd

pwd	Returns the current working directory.
pwd -p	Print the physical directory, without any symbolic links.
pwd -l	Print the value of \$PWD if it names the current working directory.

### Exit

exit : Exits the shell after printing a goodbye message.

## Externally Handled Commands

### Ls

ls	Displays the present files and folders present inside the current working directory in a space-separated format.
ls -a	Displays the present files and folders present inside the current working directory, including the ones starting with '.' in a space-separated format.
ls -l	Displays the present files and folders present inside the current working directory in a list format.

Corner cases:

- 1)If the user wants both -l and -a options together, they can type -la or -al and get a combination of the both.
- 2)If the user enters something other than 'a' or 'l' after the '-', the terminal will return 'Invalid option'

### Date

date	Displays the current date and time in IST.
date -u	Displays the current date and time in UTC.
date -r fname.txt	Displays the most recent access and modification date and time of the specified file.

Corner Cases:

- 1)If the user tried to access a file that does not exist, the terminal would display "File 'filename' could not be found!"
- 2)If the user tried to enter anything other than 'u' or 'r', the terminal would display "Invalid option"

### Cat

cat fname.txt	Concatenates and displays the contents of the specified file.
cat -n fname.txt	Concatenates and displays the contents of the specified file, while numbering each new line.
cat -E fname.txt	Concatenates and displays the contents of the specified file with displaying the start of every line by a '\$' sign.

Corner cases:

- 1)If the user entered just 'cat' : The terminal would display "Filename not provided."
- 2)If the user tried to access a file that does not exist, the terminal would display "File 'filename' does not exist!"

### Mkdir

mkdir dir_name	Creates a directory with the specified name.
mkdir -v dir_name	Describes the actions of the mkdir command as it attempts to create the directory.
mkdir -m = mode dir_name	Creates a directory with the specified name in the specified mode.

Corner Cases:

- 1)If the user does not enter the name of the directory, it returns a warning message.
- 2)If the user wants to create multiple directories at once, they can do that by writing them in the same command as well.

### Rm

rm fname.txt	Removes the specified file from the current working directory without any warning.
rm -i fname.txt	Removes the specified file from the current working directory after asking for permission.
rm -v fname.txt	Removes the specified file from the current working directory and expresses the result.

Corner Cases:

1) If the user tries to access a file that is not found, the terminal will display 'Sorry! File doesn't exist'

2) In the '-i' option, the program accepts both y and Y as acceptable outputs for going ahead with the procedure. If the user enters anything other than these, the operation will be aborted as a warning.

### Test Case

&t mkdir demo

echo hi

echo -n hello user

echo --help

pwd

ls

ls -a

cd newdir

pwd

cd ..

pwd

cd

&t cat 1.txt

cat -n 1.txt

cat -E 1.txt

date

date -u

date -r 1.txt

ls -l

ls -al

ls -la

mkdir -v trynewdemodir1

mkdir -m = rwx trynew7

rm 1.txt

rm -i 2.txt -->n

rm -i 2.txt-->y

rm -v 7.txt

exit