I have used the command execvp thus i have umoved the executable files to usr/bin, as the system call execvp finds the files in the files mentioned in the path variable PATH

External Commands:

ls

when run without any options it display all the files in the directory (but not the ones whose name begin with .)

Options: -i, -a

-i : This option displays the index number of each file in the directory.

-a: The option display all the files in the directory starting even the ones starting with .

Corner cases, bugs handled:

When the user inputs both the commands together then the command displays all the files with their index number (even the ones swhoes name start with .)

When the user inputs data in the wrongs format then the shells prompts that the arguments given are bad.

Test Cases:

Command 1: ls

Command 2: ls -i

Command 3: ls -a

Command 4: ls -i -a

Command 5: ls -i -a ..

Command 6: ls -i -a sub (where sub is a child directory of the directory in which the shell is running)

cat

Assumption: The user inputs at most two files at the same time.

When run without any options it displays the content of the single file if one file is given as the argument and it displays the content of both files if given two agruments

Options: -n, -E

-n : This option displays line number infront of every line

-E: The option display $ at the end of every line displayed on the shell

Corner cases, bugs handled:

When the user inputs both the commands together then the command displays all the lines with line number and the symbol $ at the end.

When the user inputs data in the wrong format then the shells prompts that the arguments given are bad.

When the user inputs inappropriate number of arguments then the shell prompts whether the arguments are too many or too few.

Test Cases:

Command 1: cat a.txt

Command 2: cat a.txt b.txt

Command 3: cat -n a.txt b.txt

Command 4: cat -E a.txt b.txt

Command 5: cat -n -E a.txt b.txt

Command 6: cat -n -E a.txt

(Where a.txt and b.txt are two files in the directory)

date

When run without any options it displays the time in the format (Day of the Week Date Month Year Time AM/PM TimeZone)

Options: -u, -R

-u : This option displays UTC (Coordinated Universal Time) time

-R: The option display the time in the RFC 5322 format (Day of the Week, Date Month Year Time Time Difference between UTC and current TZ)

Corner cases, bugs handled:

When the user inputs both the commands together then the command displays the UTC time in RFC 5322 format

When the user inputs data in the wrong format then the shells prompts that the arguments given are bad.

Test Cases:

Command 1: date

Command 2: date -u

Command 3: date -R

Command 4: date -u -R

rm

Assumption: The user inputs at most one files at one time

When run without any options it deletes the mentioned file

Options: -d, -v, -i

-d : This option deletes the directory mentioned if its empty

-v: The option deletes the file mentioned (not directory) and confirms the task being executed

-i: Thw option asks for the user’s permission before deleting the mentioned file

Corner cases, bugs handled:

When the user inputs all the options together, the command deletes the given directory but asks for permission and tells the user about the task being executed

When the user inputs data in the wrongs format then the shells prompts that the arguments given are bad.

When the user tries to delete a directory that is not empty, the shell prompts.

When the user tries to delete a directory without using the -d option, the shell prompts

Test Cases:

Command 1: rm a.txt

Command 2: rm -d sub

Command 3: rm -i b.txt

Command 4: rm -v c.txt

Command 5: rm -i -v d.txt

Command 6: rm -i -d -v sub2

(Where .txt files are in the directory and sub and sub2 are child directories)

mkdir

Assumption: The user inputs at most one name at one time

When run without any options it creates the directory inside the directory in which the shell is at the time of running the command

Options: -v,-p

-v: The option creates a directory and confirms the task being executed

-p: The options creates parent directories if required

Corner cases, bugs handled:

When the user inputs both the commands together then the command displays all the tasks being executed and creates all the parent directory if required

When the user inputs data in the wrongs format then the shells prompts that the arguments given are bad.

When the user inputs inappropriate number of arguments then the shell prompts whether the arguments are too many or too few.

If the user inputs the name of a directory which already exists, then shell prompts

Test Cases:

Command 1: mkdir a

Command 2: mkdir -v b

Command 3: mkdir -p c/d (creates both c and d)

Command 4: mkdir -v -p e/f/g

Internal Commands

cd

When run without any options it changes the working directory of the shell

Options: .. ,

.. : Moves the working directory of the shell to the parent of the current directory

Corner cases, bugs handled:

If the user inputs a directory which does not exists in the current directory then thh shell prompts

The user inputs bad arguments , the shell prompts with either bad address or bad argument

Test Cases:

Command 1: cd sub

Command 2: cd ..

pwd

When run the command prints the name of the working directory

Corner cases, bugs handled:

The user inputs bad arguments , the shell prompts with bad argument

When the user inputs inappropriate number of arguments then the shell prompts there are too many.

Test Cases:

Command 1: pwd

exit

When run, the command exits from the shell

echo

Assumption: The user deos not input quotation marks

When run, the command displays the arguments entered

Options: -E

-E: The option display all the arguments as it is (without using \ as an escape sequence)

Corner cases, bugs handled:

When the user inputs data in the wrongs format then the shells prompts that the arguments given are bad.

If the user inputs the name of a directory which already exists, then shell prompts

Test Cases:

Command 1: echo name

Command 2: echo name \n lastName

Command 3: echo name lastName

history

Assumption: The hisotry is recorded against all sessions

When run without any options it displays all the command run in the shell

Options: -clear,n

-clear: The option clears all the history

n (a natural number): The option displays the last n commands entered by the user

Corner cases, bugs handled:

When the user inputs data in the wrongs format then the shells prompts that the arguments given are bad.

Test Cases:

Command 1: history

Command 2: history -clear

Command 3: history 2

Command 4: history 3