

I have written a small shell which have following functions and in this i have assumed following assumption -

shell can handle 10 commands and in othercases I will print "command not found".
shell have 5 internal and 5 external commands-

| internals | externals |
|--|----------------------------|
| 'cd', 'echo', 'history', 'pwd', 'exit' | 'ls', 'cat', 'date', 'rm', |
| 'mkdir' | |

echo command -

echo the particular stiring

flags -

1. -e -> reads all the character starts with \ and prints according to there meaning

2. -n -> remove new line after string ends

exit -

successfully exit from program and close it

history command-

print the history since you have started the shell.

flags -

1. -d -> remove particular history

2. -c -> clean all the history

else it will print "command not found"

pwd -

gives you present working directory

cd -

change the directory according to given input

For all the external commands i have used execlp system call in which we pass the executable file and can pass other parameters which we can access when the main program execute. We call execlp inside the chile process which we can create using fork syscall.

ls -

list out all the files in the current directory

flags -

1. -a -> don't ignore entries starting with . (like . ..)

2. -A -> don't list implied . and ..

3. -m -> fill width with a comma separated list of entries

cat -

Concatenate FILE(s) to standard output

flags -

1. -E -> display \$ at end of each line

2. -T -> display TAB characters as ^I

date -

Display the current time

flags -

1. -u -> print or set Coordinated Universal Time (UTC)

2. -I -> output date/time

mkdir -

Create the DIRECTORY(ies), if they do not already exist

flags -

1. -p -> no error if existing, make parent directories as needed

2. -v -> print a message for each created directory

rm -

Remove the FILE

flags -

1. -f -> ignore nonexistent files and arguments, never prompt
2. -i -> prompt before every removal

Thank you