

Basic Linux/Unix Shell

I have made a Linux shell that is coded in C. It has 8 commands with 2 options each and accepts all possible arguments. It has 3 internal and 5 external commands which were implemented using fork, execv, wait and pthread system calls. It also handled various corner cases and errors.

The shell uses an infinite loop. It takes the user input as string using fgets, then splits it using strtok and stores them in an array of string called command.

Running the code in terminal:

In the terminal, write “make” to compile all the files required for the shell to run, and then write “./mainshell” to execute the code and start the shell.

Shell command options:

My shell has the following internal commands - cd, echo and pwd and external commands - date, ls, mkdir, rm, cat

1. cd:

“cd <directory>” goes to the mentioned directory

“cd ..” goes to previous directory

“cd /” goes to root directory

2. echo:

“echo <arg>” prints the arg

“echo -n” omits a new line

3. pwd:

“pwd” gives working directory

“pwd -L” gives working directory

“pwd -P” gives working directory

4. ls:

“ls” prints everything with tabspace in between

“ls -m” prints everything with commas in between

“ls -1” prints everything in a new line

5. date:

“date” prints time in IST
“date -u” prints time and date in UTC
“date -l” prints date only

6. cat:

“cat” prints statement entered
“cat -n” prints the statement with its line count
“cat -b” prints the statement with its line count excluding the count of statements that were blank

7. mkdir:

“mkdir <name>” creates directory
“mkdir -v” prints a message that directory has been created

8. rm:

“rm <name>” removes directory
“rm -i <name>” asks for confirmation before removal
“rm -d <name>” removes directory <name> if it is empty

Error handling:

1. cd:

- If directory does not exist, it gives an error message
- If any argument is passed on the second index, it gives an error message

2. echo:

- No error handling because everything written after echo is printed.

3. pwd:

- If any argument, other than the ones I’ve coded is entered after “pwd”, it still gives the working directory

4. ls:

- If any argument, other than the ones I’ve coded is entered, it gives an error message
- If any argument is passed on the second index, it gives an error message

5. date:

- If any argument, other than the ones I've coded is entered, it gives an error message
- If any argument is passed on the second index, it gives an error message

6. cat:

- If any argument, other than the ones I've coded is entered, it gives an error message
- If any argument is passed on the second index, it gives an error message

7. mkdir:

- If the first index of the command is empty, it gives an error message
- If the directory already exists, it gives an error message

8. rm:

- If the first index of the command is empty, it gives an error message
- If the file or directory does not exist, it gives an error message

Assumptions:

- Directory name should not have spaces (example: "hello 1" is invalid)
- Any command that does not exist in my code will result to error
- Home directory used is "/home/parth/Desktop/mainshell".
- Writing anything after pwd will not make a difference, it will still work as pwd only
- Writing just date command will give time in IST
- Writing any type of command ls will not show hidden files

Test cases:



```
parth@parth-virtual-machine:~$ cd Desktop
parth@parth-virtual-machine:~/Desktop$ make
gcc -o cat cat.c
gcc -o date date.c
gcc -o ls ls.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
gcc -o mainshell mainshell.c
parth@parth-virtual-machine:~/Desktop$ ./mainshell
>> echo hi hello
echo hi hello
hi hello
>> pwd
pwd
/home/parth/Desktop
>> cd ..
cd ..
/home/parth
>> cd Desktop
cd Desktop
>> pwd -L
pwd -L
/home/parth/Desktop
>> date
date
Mon Oct 31 22:00:48 2022
>> date -I
date -I
2022-10-31
>> cat
cat
hello
hello

hello1
hello1

hello2
hello2

^C
parth@parth-virtual-machine:~/Desktop$ ./mainshell
>> S
```

```
parth@parth-virtual-machine: ~/Desktop
>> pwd
pwd
/home/parth/Desktop
>> ls
ls
shellpthread  date  testmkdir.c  testdate  shell.c  t  testcat.c  testrm  date.c  testrm.c  mainshell.c  a.out  test  testdate.c  Makefile
file  cat  pthreadtest.c  mainshell  .  mkdir  testls  ls  shell.c  ls.c  rm  shell  rm.c  ..  cat.c  mkdir.c  testcat  testmkdir  testls.c  p
threadtest
>> mkdir hello
mkdir hello
>> ls -l
ls -l
shellpthread
date
testmkdir.c
testdate
shell.c
t
testcat.c
testrm
testrm
date.c
hello
testrm.c
mainshell.c
a.out
test
testdate.c
Makefile
cat
pthreadtest.c
mainshell
.
mkdir
testls
ls
shell.c
ls.c
rm
shell
rm.c
..
cat.c
mkdir.c
testcat
testmkdir
testls.c
pthreadtest

>> rm -i hello
rm -i hello
rm: remove regular empty file 'hello' (y/n)? y
hello: removed successfully
>> ls -m
ls -m
shellpthread, date, testmkdir.c, testdate, shell.c, t, testcat.c, testrm, date.c, testrm.c, mainshell.c, a.out, test, testdate.c, Makefile, cat, pthreadtest.c, mainshell, ., mkdir,
r, testls, ls, shell.c, ls.c, rm, shell, rm.c, .., cat.c, mkdir.c, testcat, testmkdir, testls.c, pthreadtest,
```