

ASSIGNMENT-1

BASIC LINUX/UNIX SHELL

AAJAY AYYAPPAN DEVARAJ [2021001]

Implemented Basic Linux Shell using C and the following C libraries: -

- `<stdio.h>`
- `<stdlib.h>`
- `<string.h>`
- `<unistd.h>`
- `<pthread.h>`
- `<dirent.h>`
- `<time.h>`
- `<sys/types.h>`
- `<sys/stat.h>`
- `<limits.h>`

The following Shell supports 3 internal commands:

1. `cd`
2. `echo`
3. `pwd`

And the 5 external commands:

1. `ls`
2. `cat`
3. `rm`
4. `date`
5. `mkdir`

The shell handles and executes the external commands in Two ways:

1. Using `fork()` / `execv()` system calls
2. `pthread_create()` / `system()`

COMPILATION USING MAKEFILE:

To Compile the C files-

Run "make"

```
● (base) aajaydevaraj@Aajays-MacBook-Pro Shell % make
gcc -o shell shell.c
gcc -o ls ls.c
gcc -o cat cat.c
gcc -o date date.c
gcc -o mkdir mkdir.c
gcc -o rm rm.c
○ (base) aajaydevaraj@Aajays-MacBook-Pro Shell %
```

To run the shell, enter "make run"

```
○ (base) aajaydevaraj@Aajays-MacBook-Pro Shell % make run
./shell
orion>
```

To delete previously compiled files,

Run "make clean"

```
● (base) aajaydevaraj@Aajays-MacBook-Pro Shell % make clean
rm -f shell ls cat date rm mkdir
○ (base) aajaydevaraj@Aajays-MacBook-Pro Shell %
```

(These commands can be run in the directory of the folder containing the shell files and the Makefile to successfully compile and run the program)

DOCUMENTATION ON IMPLEMENTED SHELL COMMANDS: -

EXTERNAL COMMANDS:

ls:

list directory contents

ls [*OPTION*] ... [*FILE*] ...

supports the options:

1. -a : do not ignore entries starting with . and ..
2. -i : print the index of each file

Errors Handled:

1. Invalid option
2. Invalid number of arguments (valid format must be of the form 'ls [-i] [-a] [directory]')
3. Provided file or directory does not exist

cat:

concatenate files and print on the standard output

cat [*OPTION*] ... [*FILE*] ...

supports the options:

1. -E : display \$ at the end of each line
2. -n : number all output lines

Errors Handled:

1. Invalid option
2. No such file or directory exists

date:

print or set the system date and time

date [*OPTION*] ..

supports the options:

1. -u : print coordinated universal time
2. -R : output date and time in RFC 5322 format.

Errors Handled:

1. Invalid option
2. Invalid number of arguments

rm:

remove files or directories

rm [*OPTION*] ... [*FILE*] ...

supports the options:

1. -i : prompt before every removal
2. -v : explain what is being done
3. -d : remove empty directories

Errors Handled:

1. Invalid option
2. No such file or directory
3. Directory is not empty , thus cannot be removed

mkdir:

make directories

mkdir [*OPTION*] ... *DIRECTORY*...

supports the options:

1. -v : print a message before each created directory
2. -p : make parent directories as needed , no error if existing

Errors Handled:

1. Invalid option
2. File already exists, cannot create directory

SWITCHING BETWEEN FORKING AND THREADING TO RUN COMMANDS:

Adding '&t' at the end of each command line input will make the shell switch to thread-based execution rather than running the commands through processes created by fork().

Example:-

- ls -i &t
- mkdir -v a &t
-

INTERNAL COMMANDS:

echo:

display a line of text

echo [-n] [-e] "STRING"

supports the options:

1. -n : do not output the trailing new line
2. -e : enable interpretation of backslash escapes

Errors Handled:

1. Invalid option
2. No arguments provided

cd:

change the working directory

```
cd [-L|-P] [directory]
```

```
cd -
```

supports the options:

1. -P : handles operand physically
2. - : moves to the preceding directory

Errors Handled:

1. Invalid option
2. No such file or directory exists

pwd:

print name of current/working directory

```
pwd [OPTION]...
```

supports the options:

1. -P : print path without symlinks
2. -L : print path with symlinks

Errors Handled:

1. Invalid option
2. Options -P and -L are mutually exclusive

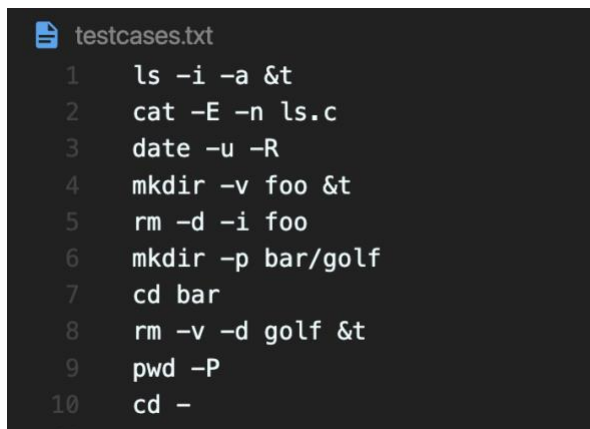
exit:

to exit out of the running shell

`'exit'`

NOTE:- ERRORS BASED ON INVALID COMMAND INPUT ARE ALSO HANDLED

TEST CASES:



```
testcases.txt
1  ls -i -a &t
2  cat -E -n ls.c
3  date -u -R
4  mkdir -v foo &t
5  rm -d -i foo
6  mkdir -p bar/golf
7  cd bar
8  rm -v -d golf &t
9  pwd -P
10 cd -
```

Some examples of commands that can be run:-

```
ls -i -a &t
cat -E -n ls.c
date -u -R
mkdir -v foo &t
rm -d -i foo
mkdir -p bar/golf
cd bar
rm -v -d golf &t
pwd -P
cd -
```

any of the listed commands in this document, with the specified options, can be run. With `'&t'` at the end to switch between `fork()` and `Pthread_create()` methods of execution.

REFERENCES USED:-

1. Linux Manual Pages [for information on commands and their options]

<https://man7.org/linux/man-pages/>