

# Assignment 1 - Q2

Files included:

1. Makefile
2. Main C program with the internal and external commands.
3. 5 files for the external commands (ls, cat, date, rm, mkdir)
4. Some test files and directories
5. History file
6. Help files

All commands (external and internal) except echo(which did not work in my system) has --help flag.

In external commands, fork() is used. In the child process, execvp() is used and the arguments are passed to it to an external file which implements the external command and then exit() is used to exit the child process. In the parent process, wait() is used to make the parent process wait for the completion of the child process.

Internal commands

1. Cd

System calls used:

l. chdir() has been used to change directory.

Error handling:

If the directory does not exist, then the shell gives an error message.

Flags:

cd . and cd .. have been handled.

--help flag has been implemented.

2. Echo:

System calls used:

No system calls are used, just the output is printed to the console.

Error Handling:

There is no error handling required for this command for the flags implemented.

Flags:

-n flag had been implemented which prints the output without going to the next line.

3. History:

System calls used:

Read, write and open are used to store all the commands into a history.txt file.

Error handling:

There was no need for error handling.

Flags:

- I. --help flag is implemented.

#### 4. Pwd:

System calls used:

The home path for the program directory has been hardcoded.

Error handling:

There was no need for error handling.

Flags:

- P flag is implemented which does not print the symlinks.
- help flag is implemented.

#### 5. Exit:

System calls used:

No system calls used. The program takes input until the input is "exit".

Error handling:

No need for error handling.

Flags:

Not applicable

### External Commands

#### 1. LS:

System class used:

Opendir() and readdir() are used to get the directory given and then it loops over all the files and folders of the directory and prints their names.

Error Handling:

If the directory does not exist, then the console throws an error.

Flags:

- a flag is implemented which prints all the contents that start with ".".
- l flag is implemented which prints all the contents on different lines.
- help flag is implemented.

#### 2. Cat:

System calls used:

open() and read() are used to open and read the contents of the file.

Error handling:

If the file does not exist, then the console throws an error.

Flags:

- n flag is implemented, which prints the line number with each line in the output.
- E flag is implemented, which prints a "\$" after each line in the output.
- help flag is implemented.

## 3. Date:

System calls used:

The header file time.h is used. localtime() is used to get the local time and date.

Error handling:

No error handling needed for the flags implemented.

Flags:

--help is implemented.

## 4. Rm:

System calls used:

remove() is used to delete a file.

Error handling:

If the file does not exist, then the console throws an error.

Flags:

-f flag is implemented, which does not give an error even if the file does not exist.

-i flag is implemented, which asks for confirmation to delete the file.

--help is implemented.

## 5. Mkdir:

System calls used:

mkdir() is used to create a directory.

Error Handling:

If the directory already exists, then it gives an error message.

Flags:

-v flag is implemented, which gives a message that a directory has been created.

-p flag is implemented, which does not give an error if the directory already

exists.

--help is implemented.

## Test cases (with screenshots)

## LS

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ ls -l test_folder
OS
file
fol1
test.txt
folder

/home/hari/OS/Assignments/Assignment 1/Q2/ ls -a test_folder
OS file fol1 . test.txt .. folder
/home/hari/OS/Assignments/Assignment 1/Q2/ ls test_folder
OS file fol1 test.txt folder
/home/hari/OS/Assignments/Assignment 1/Q2/ █

```

## Cat

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ cat callee.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    long n1 = strtol(argv[1], 0, 0);
    long n2 = strtol(argv[2], 0, 0);
    printf("%s\n", argv[1]);
    printf("%ld\n", n1+n2);
    return 0;
}

/home/hari/OS/Assignments/Assignment 1/Q2/ cat -n callee.c
1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 int main(int argc, char *argv[]){
5     long n1 = strtol(argv[1], 0, 0);
6     long n2 = strtol(argv[2], 0, 0);
7     printf("%s\n", argv[1]);
8     printf("%ld\n", n1+n2);
9     return 0;
10 }

/home/hari/OS/Assignments/Assignment 1/Q2/ cat -E callee.c
#include <unistd.h>$
#include <stdio.h>$
#include <stdlib.h>$
int main(int argc, char *argv[]){$
    long n1 = strtol(argv[1], 0, 0);$
    long n2 = strtol(argv[2], 0, 0);$
    printf("%s\n", argv[1]);$
    printf("%ld\n", n1+n2);$
    return 0;$
}$

/home/hari/OS/Assignments/Assignment 1/Q2/ █

```

## Date

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ date
now: 2020-09-30 20:58:16 IST
/home/hari/OS/Assignments/Assignment 1/Q2/

```

## Rm

```

hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt date mkdir.c cat.c cat_help.txt rm test_folder NEWFOLDER_MESSAGE function ls.c hi
story.txt NEWFOLDER mkdir mkdir_help.txt FILETOREMOVE.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller c
ommands.c caller.c ls_help.txt function2 date.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ rm -i FILETOREMOVE.txt
rm: cannot remove 'FILETOREMOVE.txt': No such file or directory
rm: remove regular file 'FILETOREMOVE.txt'?
y
/home/hari/OS/Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt date mkdir.c cat.c cat_help.txt rm test_folder NEWFOLDER_MESSAGE function ls.c hi
story.txt NEWFOLDER mkdir mkdir_help.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller commands.c caller.
c ls_help.txt function2 date.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ rm FILETOREMOVE.txt
rm: cannot remove 'FILETOREMOVE.txt': No such file or directory
/home/hari/OS/Assignments/Assignment 1/Q2/ rm -f FILETOREMOVE.txt
/home/hari/OS/Assignments/Assignment 1/Q2/

```

(FILETOREMOVE.txt is removed)

## Mkdir

```

hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/ Google Chrome Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt date mkdir.c cat.c cat_help.txt rm test_folder function ls.c history.txt mkdir
mkdir_help.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller commands.c caller.c ls_help.txt function2 d
ate.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ mkdir NEWFOLDER
/home/hari/OS/Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt date mkdir.c cat.c cat_help.txt rm test_folder function ls.c history.txt NEWFOLDE
R mkdir mkdir_help.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller commands.c caller.c ls_help.txt f
unction2 date.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ mkdir -p NEWFOLDER
/home/hari/OS/Assignments/Assignment 1/Q2/ mkdir NEWFOLDER
mkdir: cannot create directory 'NEWFOLDER': File exists
/home/hari/OS/Assignments/Assignment 1/Q2/ mkdir -v NEWFOLDER_MESSAGE
mkdir: created directory 'NEWFOLDER_MESSAGE'
/home/hari/OS/Assignments/Assignment 1/Q2/

```

## Echo

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ echo output
output
/home/hari/OS/Assignments/Assignment 1/Q2/ echo -n withoutline
withoutline/home/hari/OS/Assignments/Assignment 1/Q2/
/home/hari/OS/Assignments/Assignment 1/Q2/

```

## History

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt date mkdir.c cat.c cat_help.txt rm test_folder NEWFOLDER_MESSAGE function ls.c hl
story.txt NEWFOLDER mkdir mkdir_help.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller commands.c caller.
c ls_help.txt function2 date.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ install
install: command not found
/home/hari/OS/Assignments/Assignment 1/Q2/ rm date
/home/hari/OS/Assignments/Assignment 1/Q2/ ls .
history_help.txt date_help.txt ls rm_help.txt mkdir.c cat.c cat_help.txt rm test_folder NEWFOLDER_MESSAGE function ls.c history.tx
t NEWFOLDER mkdir mkdir_help.txt callee cd_help.txt cat callee.c back1 Makefile shell rm.c caller commands.c caller.c ls_
help.txt function2 date.c pwd_help.txt
/home/hari/OS/Assignments/Assignment 1/Q2/ history
1 ls .
2 install
3 rm date
4 ls .
5 history

/home/hari/OS/Assignments/Assignment 1/Q2/

```

## PWD

```

hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ cd test_folder
bash: cd: test_folder: No such file or directory
/home/hari/OS/Assignments/Assignment 1/Q2/ cd NEWFOLDER
/home/hari/OS/Assignments/Assignment 1/Q2/NEWFOLDER/ pwd
/home/hari/OS/Assignments/Assignment 1/Q2/NEWFOLDER/
/home/hari/OS/Assignments/Assignment 1/Q2/NEWFOLDER/

```

## Exit

```
hari@hari-Vostro-5481: ~/OS/Assignments/Assignment 1/Q2$ make commands
gcc date.c -o date
gcc ls.c -o ls
gcc rm.c -o rm
gcc mkdir.c -o mkdir
gcc cat.c -o cat
gcc commands.c -o shell
./shell

/home/hari/OS/Assignments/Assignment 1/Q2/ exit
hari@hari-Vostro-5481:~/OS/Assignments/Assignment 1/Q2$
```