

# Question 2

## Working of the Shell

For external commands shell uses exec family of system calls. For internal commands it uses .h files which are compiled with it.

## Error handling

The following errors are being handled by different commands-:

1. Invalid Flags
2. Invalid files
3. Opening non existent folders
4. Opening non existent files
5. Opening files shell does not have permission for.
6. Opening folders shell does not have permission for.
7. Reading and writing on non existent files.

## ASSUMPTION

1. PLACE THE FILE IN HOME FOLDER
2. CHANGE NAME FROM CHINTAN TO YOUR USER.
3. NO TWO FLAGS ARE USED TOGETHER.
4. THERE ARE NO SPACES BETWEEN NAMES FOR COMMANDS EXCEPT MKDIR.

# Commands and flags

- 'Cd',
  - Flags
  - --help : shows help
  - -L: resolves symbolic links
- 'echo',
  - Flags
  - -n remove newline from end
  - --help : shows help
- 'History',
  - Flags
  - --help : shows help
  - -a : appends history to file
- 'Pwd'
  - Flags
  - --help : shows help
  - -L : process symbolic links
- "Exit"
  - Flags.
  - [n] exit with this code
  - --help
- 'Ls',

- Flags
  - -1 one file per line
  - -Q all files in quotes
- 'cat',
  - Flags
  - -n no at start of each line
  - -E dollar at end of each line
- 'Date'
  - Flags,
  - -R shows date in rfc 5322 format
  - --help shows help
- 'Rm'
  - Flags
  - -f : force remove
  - -i : interactive remove
- 'Mkdir'.
  - Flags
  - --help : shows help
  - -v : verbose mode

# Test cases

The image displays a terminal window and two Google Docs screenshots. The terminal window shows the output of a program named 'boomerop' with various flags and options. The Google Docs screenshots show the test cases for the program, organized into sections for 'cat', 'rm', and 'mkdir'.

**Terminal Window Output:**

```
DESCRIPTION
Write arguments to the standard output.

Display the ARGs, separated by a single space character and followed by a
newline, on the standard output.

Options:
-n          do not append a newline

boomerop>>ls
ln child process
mkdir.help  history.txt  ls          Makefile    cd.help
as         history.help shell.c temp    cd.h        pwd.help    extt.help   temp.c
date.c     mkdir.c echo.help  date.help   echo.h      cat.c      rm          history.h
date       mkdir    shell     ls.c
cat        rm.c
boomerop>>rm -t temp
ln child process
Do you want to remove (y/n) temp
n
boomerop>>rm -f temp
ln child process
boomerop>>rm temp
ln child process
rm cannot delete temp
boomerop>>mkdir as
ln child process
mkdir cannot make 'as' dir
boomerop>>ls
ln child process
mkdir.help  history.txt  ls          Makefile    cd.help
as         history.help shell.c cd.h    cd.h        pwd.help    extt.help   temp.c
date.c     mkdir.c echo.help  date.help   echo.h      cat.c      rm          history.h
date       mkdir    shell     ls.c
cat        rm.c
boomerop>>mkdir asa
ln child process
boomerop>>ls
ln child process
mkdir.help  history.txt  ls          Makefile    cd.help
as         history.help shell.c cd.h    cd.h        pwd.help    extt.help   temp.c
date.c     mkdir.c echo.help  date.help   echo.h      cat.c      rm          history.h
date       mkdir    shell     ls.c
cat        rm.c
asa        rm.c
boomerop>>exit 2
exiting now...
chintan@chintan-Lenovo-Ideapad-510-15IKB:~/part2$ echo $?
2
chintan@chintan-Lenovo-Ideapad-510-15IKB:~/part2$
```

**Google Docs Screenshot 1 (Top):**

- 'cat':
  - o Flags
  - o -n no at start of each line
  - o -E dollar at end of each line
- 'Date':
  - o Flags
  - o -R shows date in rfc 5322 format

**Google Docs Screenshot 2 (Bottom):**

- 'cat':
  - o -help shows help
- 'rm':
  - o Flags
  - o -f: force remove
  - o -i: interactive remove
- 'mkdir':
  - o Flags
  - o -help : shows help
  - o -v : verbose mode

Test cases





















