Write up

For compilation:
use the "make" command to compile the files.

Usage:
use the "./shell" command to run the shell.

Assumption and Errors:
1. If the user enters without typing any command, the program goes into an infinite loop.
2. File not found an error in the cat, rm.
3. If the user changes the current directory, then none of the external commands will work because the program searches for binary in the current folder.
4. Once the "history -c" command is used, "history" cannot be used to see history as history would clear by history clear command.
5. If the user types any command with leading space before the command, then the leading space would be ignored.

    E.g.: '   echo hello world.'

Commands and systems call used:
I have used the execvp() function from the execl() family system call to call the binaries of external commands. I have also used fork() and wait() system call to create child processes and to wait for the child program to terminate.

Command-line option :
1.help command for documentation
Internal Commands :
2.cd = to change the current directory
3.history = show command history
    -c : clear history

4.echo = print arguments

   -n : do not output the trailing newline

5.pwd = print working directory

6.exit = exit the shell

External Commands:

7.ls = list directory contents

   -a : show hidden files

8.cat = open files

   -b : show line number

9.date = show date

   -u : date in UTC

10.rm = remove file

   -v : verbose

11.mkdir = make directories

   -v : verbose

   -m : make directory with permission

Other functions used:

1. mkdir - mkdir()
2. pwd - getcwd()
3. cd - chdir()
4. cat - fgetc()
5. ls - scandir()
6. date - time()
7. rm - remove()