Exercise 1.2 Documentation

Basic Linux/Unix shell

System:

- The attached code is written in 6 separate c files.
- It functions as a Basic Linux Shell which implements the following commands:
  - Internal:
    - cd (Change the shell working directory.)
      - -P (use the physical directory structure without following symbolic links: resolve symbolic links in DIR before processing instances of `..')
      - -L (force symbolic links to be followed: resolve symbolic links in DIR after processing instances of `..')
      - --help
    - echo (display a line of text)
      - -n (do not output the trailing newline)
      - -E (disable interpretation of backslash escapes)
      - --help
    - history (GNU History Library)
      - -a
      - -c
      - --help
    - pwd (print name of current/working directory)
      - -P (avoid all symlinks)
      - -L (use PWD from environment, even if it contains symlinks)
      - --help
    - exit (cause normal process termination)
  - External:
    - ls (list directory contents)
      - -a (do not ignore entries starting with .)
      - -l (use a long listing format)

- cat (concatenate files and print on the standard output)
  - -E (display $ at end of each line)
  - -T(display $ at end of each line)
- date (print or set the system date and time)
  - -R (output date and time in RFC 5322 format)
  - -u (print or set Coordinated Universal Time (UTC))
- rm (remove files or directories)
  - -d (remove empty directories)
  - -v (explain what is being done)
- mkdir (make directories)
  - -v (print a message for each created directory)
  - -m (set file mode (as in chmod), not a=rwx – umask)

- The major errors handled and assumptions made throughout the program were mainly the ones:
  - Wrong commands
  - Wrong command line option
  - Directory doesn’t exist
  - Can’t delete directories by default
  - Directory already exists
  - wrong mode for creation of directory
  - file not present
  - The user would always enter a command (even though wrong)
  - The user would always enter the complete path (even though wrong)
  - and so on…

- Test Case:
  - ls
  - ls -l
  - ls -a
  - ls -a path
  - cat file1

- cat file1 file2
- cat -E file1
- cat -T file2
- date
- date -R
- date -u
- rm file1
- rm file1 file2
- rm -d directory1
- rm -v file1
- mkdir a
- mkdir c b
- mkdir -v d
- mkdir -m=0100 e
- cd
- cd path
- cd -P path
- cd -L path
- cd –help
- echo text
- echo -n text
- echo -E text
- echo –help
- history
- history -a 10
- history
- history -c
- history
- history –help
- pwd
- pwd -P
- pwd -L

- exit