Charles L. Heculi 12/7/2022

WD -317

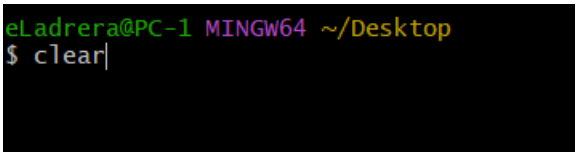
Observation For Command line Interface – Hands On 1

1. **cat Command -** I read, concatenated, or wrote file contents to the standard output using the cat command.

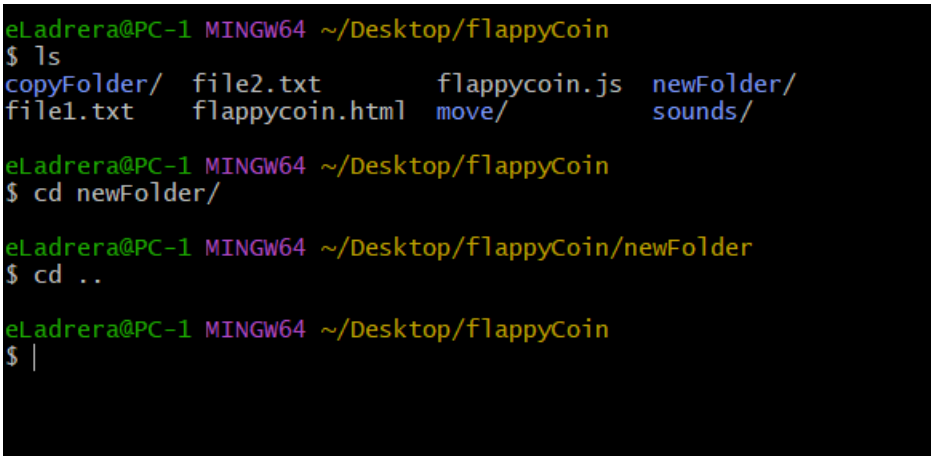
2. **mkdir Command** - I used mkdir command because it is useful when it comes to creating new directories in the file system.

3. **ls Command -** I ran the ls command from here to see the files and folders inside the flappy coin folder since I wanted to see them.

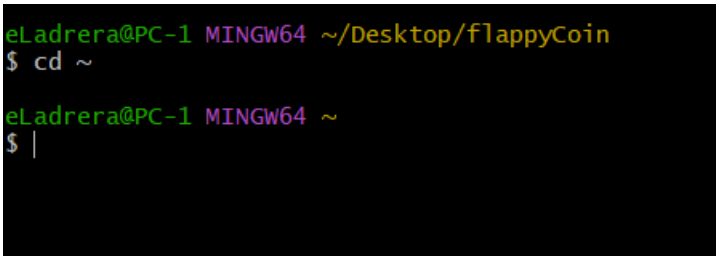
4. **clear Command** – I used clear command to clean the other command.



5. **cd .. Command** - In order to exit the sample folder, I typed the cd command. allows me to descend one level within the folder, returning the file path to the flappycoin folder as a result.



6. **cd ~ Command** - Using the cd command, I may return to my home directory.



7. **chmod -r Command** - So I may set the access controls to read only by using the chmod -r command for style.css. The chmod command is used to change the permissions. It is then followed by the filename and the -r (read-only) option.

```
eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin
$ chmod -r style.css
```

8. **mv Command** - To move directories or files from one place in the file system to another, I used the mv (move) command.

```
eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin
$ ls
flappycoin.html  flappycoin.js  move/  newFolder/  sounds/

eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin
$ mv flappycoin.html move/

eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin
$
```

9. **pwd Command** - To report the absolute path of the current working directory, I used the pwd (print working directory) command.

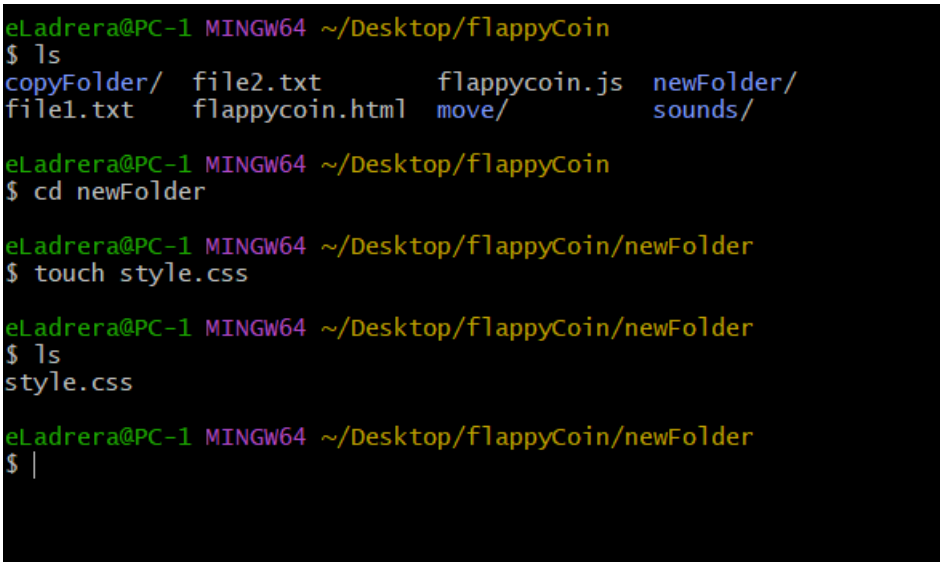
```
eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin/copyFolder
$ pwd
/c/Users/ericp/Desktop/flappyCoin/copyFolder

eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin/copyFolder
$
```

10. **rmdir Command** - to remove the sample folder, I used rmdir command to remove the New folder file

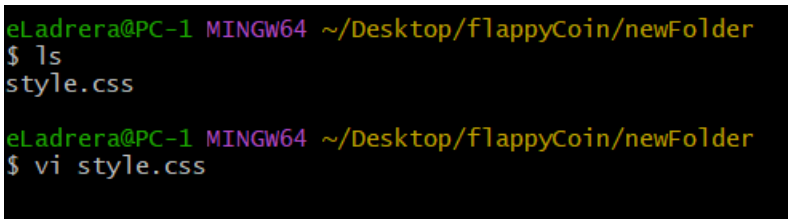
```
eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin/move
$ mkdir newFolder

eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin/move
$ rmdir newFolder

eLadrera@PC-1 MINGW64 ~/Desktop/flappyCoin/move
$ S
```

11. **cp Command** - To copy files or directories within the file system, I used the cp command.

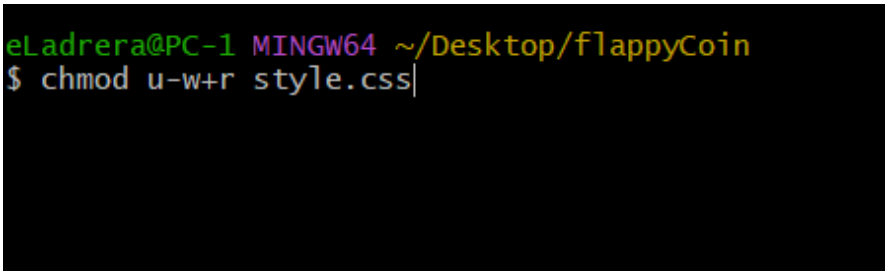
12. **touch Command** - You can update the time stamp on existing files or folders as well as create new, empty files using the touch command. When you execute the touch command on an existing file, the command will only change the time stamp. This command will just create the files if they don't already exist.



13. **vi Command** - The command vi style.css allows me to utilize the bash editor since I wanted to edit the style.css



14. **chmod u-w+r Command** - So, by using chmod u-w+r style.css, you may prevent users from writing to the file. but with a read-only authorization only. Again, chmod will provide permission, where u stands for user, -w is to prevent writing by the user, and +r is to grant the user read-only access.



15. help Command – I used help command to provide information on different command line.

```
eLadrera@PC-1 MINGW64 ~/Desktop
$ help
GNU bash, version 5.1.16(1)-release (x86_64-pc-msys)
These shell commands are defined internally.  Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

 job_spec [&]                            history [-c] [-d offset] [n] or hist>
 (( expression ))                        if COMMANDS; then COMMANDS; [ elif C>
 . filename [arguments]                  jobs [-lnprs] [jobspec ...] or jobs >
 :                                       kill [-s sigspec | -n signum | -sigs>
 [ arg... ]                              let arg [arg ...]
 [[ expression ]]                        local [option] name[=value] ...
 alias [-p] [name[=value] ... ]          logout [n]
 bg [job_spec ...]                       mapfile [-d delim] [-n count] [-O or>
 bind [-lpsvPSVX] [-m keymap] [-f file>  popd [-n] [+N | -N]
 break [n]                               printf [-v var] format [arguments]
 builtin [shell-builtin [arg ...]]       pushd [-n] [+N | -N | dir]
 caller [expr]                           pwd [-LPW]
 case WORD in [PATTERN [| PATTERN]...)>  read [-ers] [-a array] [-d delim] [->
 cd [-L|[-P [-e]] [-@]] [dir]            readarray [-d delim] [-n count] [-O >
 command [-pVv] command [arg ...]        readonly [-aAf] [name[=value] ...] o>
 compgen [-abcdefgjksuv] [-o option] [>  return [n]
 complete [-abcdefgjksuv] [-pr] [-DEI]>  select NAME [in WORDS ... ;] do COMM>
 compopt [-o|+o option] [-DEI] [name .>  set [-abefhkmnptuvxBCHP] [-o option->
 continue [n]                            shift [n]
 coproc [NAME] command [redirections]    shopt [-pqsu] [-o] [optname ...]
 declare [-aAfFgiIlnrtux] [-p] [name[=>  source filename [arguments]
 dirs [-clpv] [+N] [-N]                  suspend [-f]
 disown [-h] [-ar] [jobspec ... | pid >  test [expr]
 echo [-neE] [arg ...]                   time [-p] pipeline
 enable [-a] [-dnps] [-f filename] [na>  times
 eval [arg ...]                          trap [-lp] [[arg] signal_spec ...]
 exec [-cl] [-a name] [command [argume>  true
 exit [n]                                type [-afptP] name [name ...]
 export [-fn] [name[=value] ...] or ex>  typeset [-aAfFgiIlnrtux] [-p] name[=>
 false                                   ulimit [-SHabcdefiklmnpqrstuvxPT] [l>
 fc [-e ename] [-lnr] [first] [last] o>  umask [-p] [-S] [mode]
 fg [job_spec]                           unalias [-a] name [name ...]
 for NAME [in WORDS ... ] ; do COMMAND>  unset [-f] [-v] [-n] [name ...]
 for (( exp1; exp2; exp3 )); do COMMAN>  until COMMANDS; do COMMANDS; done
 function name { COMMANDS ; } or name >  variables - Names and meanings of so>
 getopts optstring name [arg ...]        wait [-fn] [-p var] [id ...]
 hash [-lr] [-p pathname] [-dt] [name >  while COMMANDS; do COMMANDS; done
 help [-dms] [pattern ...]               { COMMANDS ; }
```