Assignment 2
COP 290
By Sourav Bansal
2018CS50421

The command to be used to run the Assignment is **`make`** command. It starts the shell. The shell has the prompt for entering the command which displays the address of the current directory and “shell>” prompt.

The type of commands handled:

1) `builtin commands:` **`pwd, mkdir, rmdir, cd, exit`**
`These commands use the standard functions in` **`C language`** `which are` **`getcwd(), mkdir(), rmdir(), chdir(), exit()`** `respectively.`

2) Any other command typed with ‘**`./`**’ followed by the name of the file executes the file. For example- **`./a.out`** will execute the file **`./a.out`**.

3) Input can be taken from a file using ‘**`<`**’ followed by the filename. Similarly, the output can be redirected to a file using ‘**`>`**’ followed by the filename. Supported commands:
`./a.out < inp.txt > out.txt , ./a.out < inp.txt , ./a.out>out.txt`
`./a.out<inp.txt>out.txt , ./a.out<inp.txt`
The programs runs independent of space around **`> or <`** signs.

4) The output of one command can be redirected to another by using **“|”** symbol. This has been done using piping. Supported commands are:
`./a.out|./b.out , ./a.out | ./b.out`

Piping along with redirection to and from a file are also supported:
`./a.out < inp.txt | ./b.out > out.txt , ./a.out|./b.out>out.txt`
These commands work independent of the space around **|,<,>** characters.
Piping is also supported along with the pwd command. It redirect the output of builtin pwd command to another command, ie, **`pwd | ./a.out > out.txt`**

5) Other commands runs with the **`execvp()`** command in C language. These include the **`ls`** command to display file, **`clear`** to clear the shell screen, **`cat`** to access the contents of the file, **`echo`** to display message on the shell screen and others.

Some messages have been displayed as a proof that the commands are being run using the function in the C program, not by the original terminal.