

Linux command line

Open linux terminal

- Press the "windows key" or "command key"
- Type "terminal"
- Open the Terminal Emulator

Help page - man

- All linux commands can be looked up using the manual command.
- \$ man commandname

Linux file structure

Print your working directory

\$ pwd

List all files and directories (folders) in your working directory

\$ Is

Change to a different directory

\$ cd directory

Go to your home directory (home/student)

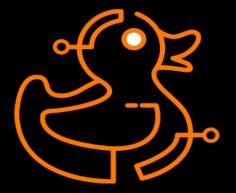
• \$ cd ~

Go back to parent directory

• \$ cd ..

Example output:





Creating and editing files

Make a directory

• \$ mkdir directoryname

Create a file in that directory

- Change directory:
 - \$ cd directoryname
- Then create the file using:
 - \$ touch filename.txt
- (Or \$ touch directoryname/filename.txt if we don't want to move into the directory)

Place text in the file

- \$ echo Hello World > filename.txt
- To add more text to the file:
- \$ echo Hello World 2 >> filename.txt
- Note that > will overwrite anything in the file whereas >> adds to the existing content.

View file

- \$ less filename.txt
- Press "q" to exit less

Open file (view and edit)

\$ subl filename.txt

Remove file

- \$ rm -i filename
- -i does a check before removal (type y when check appears)

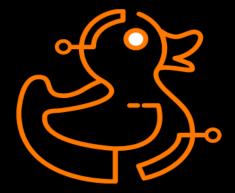
Remove directory

- \$ rm -di directoryname
- -d means remove directory

Example output:

```
(student@ kali) - [-/Documents]
$ mkdir directoryname
(student@ kali) - [-/Documents/directoryname]
$ touch filename.txt
(student@ kali) - [-/Documents/directoryname]
$ echo Hello World > filename.txt
(student@ kali) - [-/Documents/directoryname]
$ echo Hello World 2 >> filename.txt
(student@ kali) - [-/Documents/directoryname]
$ less filename.txt
(student@ kali) - [-/Documents/directoryname]
$ subl filename.txt
(student@ kali) - [-/Documents/directoryname]
$ subl filename.txt
(student@ kali) - [-/Documents/directoryname]
$ filename.txt
(student@ kali) - [-/Documents/directoryname]
$ rm - i filename.txt
rm: remove regular file 'filename.txt'? y
(student@ kali) - [-/Documents/directoryname]
$ ls
(student@ kali) - [-/Documents/directoryname]
$ cd ..
(student@ kali) - [-/Documents/directoryname]
rm: remove directoryname/'? y
```





More file commands

Copy

• \$ cp filename1.txt filename2.txt

Move

- Move filename2.txt to filename3.txt. Not, unlike cp, filename 2 will no longer exist.
- \$ mv filename2.txt filename3.txt
- (We can also give file paths e.g. \$ mv filename2.txt ~/Documents/filename4.txt)

Concatenate

- \$ cat filename1.txt filename3.txt
- Joins two files together and then prints them to the screen.
- We can easily send the output to a new file using the ">" operator:
- \$ cat filename1.txt filename3.txt > filename4.txt

Search

- \$ grep -i "Hello" filename4.txt
- This searches for the word Hello in filename4.txt. -i means ignore case.
- To search through all files that begin with file and end with .txt:
- \$ grep -i "Hello" file*.txt
- The output will indicate which file each result comes from.

Word count

- \$ wc filename1.txt
- Prints the number of lines, word, and bytes in the file

Example output:

```
└s cp file1.txt file2.txt
 (student⊕ kali) - [~/Documents/directory]

$ mv file2.txt file3.txt

(student⊕ kali) - [~/Documents/directory]

$ ls
file1.txt file3.txt

(student% kali) - [~/Documents/directory]

s cat file1.txt file3.txt

Hello World

Hello World 2

Hello World

Hello World
 (student⊕ kali)-[~/Documents/directory]

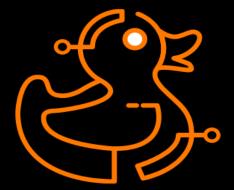
$ cat file1.txt file3.txt > file4.txt

(student⊕ kali)-[~/Documents/directory]

$ grep -i "Hello" file4.txt

Hello World

Hello World 2
             World 2
              World
             World 2
    — (student⊕ kali)-[~/Documents/directory]
-$ grep -i "Hello" file*.txt
                                   World
                                   World 2
World
                                   World 2
                                   World
                                   World 2
World
                                   World
        (student⊛ kali) - [~/Documents/directory]
                file1.txt
```



Combining commands

Piping

- We can pipe "|" the output from one command to another command.
- For example, if I want to count how many words are in all my files all of the them, I could first run \$ cat file*.txt and then pipe the output to wc:
- \$ cat file* | wc

Multiple commands

- Use && to run a second command
 - \$ program1 && program2 && program3
 - Note this will only run the next program in sequence if the previous one runs.
- If we want the next program to run, <u>no matter the outcome</u> of the previous one then we can use ":"
 - \$ program1; program2; program3

Example output: Note that in "\$ cat file2.txt && wc file1.txt" we never get a word count.

Variables

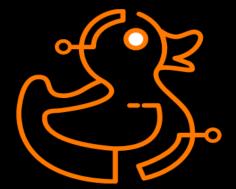
Create variable

- \$ export *variable=4*
- This variable only exists in this terminal

View variable

• \$ echo \$variable





Tips

Tab completion

- If you begin typing a file or directory name press tab to autocomplete it.
- If there is only one option for autocompletion, it will choose this option
- If there are multiple options then it will display all options

Previously typed commands

• To re-run the same command again press the up arrows on your keyboard to load previously typed commands and press enter to execute.

Run processes in the background

- Type process with "&" at the end:
- \$ firefox &

Clear terminal screen

\$ clear

