

# Reading 01

## ★ Book: The Linux Command Line ★

### ★ Chapter 7: I/O Redirection

#### • Redirecting Output

- "ls > file-list.txt" → writes the output of ls to file-list.txt
  - ↳ If you run this again file-list.txt will be overwritten.
- "ls >> file-list.txt" → the output is appended to file-list.txt

#### • Redirecting Input

- "sort < file-list.txt" → the sort command processes the contents of file-list.txt
- "sort < file-list.txt > sorted-file-list.txt" → input and output redirected here
  - ↳ order does not matter

#### • Pipelines

- Pipelines are formed by connecting multiple commands together

↳ the standard output of one command is fed into the standard input of another.

↳ Ex: "ls -l | less"

↳ the output of the ls command is fed into less.

#### • Filters

- filters take standard input and perform an operation upon it and send the results to standard output.

## • Example of pipeline!

↳ "cat unsorted\_list\_with\_dupes.txt | sort | uniq | pr | lpr"

↳ First cat sends the list into sort which sorts it

↳ This is then fed into uniq which removes any duplicates.

↳ Next pr and lpr are used to paginate and print the list.

★ REMEMBER each command outputs to standard output, but then the output is piped into the standard input for the next command.

## ★ Chapter 8: Expansion

• With expansion, we type something and it is expanded into something else before the shell acts upon it.

• Arithmetic Expansion: allows shell prompt to be used as calculator. (Only with integers)

↳ "echo \$(expression)"

• Brace Expansion

↳ useful for making directories, you can add text in front and after the braces as well.

↳ "mkdir {2017..2019}-{01..12}" creates:

|         |         |         |
|---------|---------|---------|
| 2017-01 | 2018-01 | 2019-01 |
| :       | :       | :       |
| 2017-12 | 2018-12 | 2019-12 |

• Parameter Expansion

• to see a list of available variables: "printenv | less"

• to see contents of USER variable: "echo \$USER"

## • Command Substitution

• "ls -l \$(which cp)"

↳ passing the results of which cp as an argument to the ls command.

## • Quoting

• The shell provides a mechanism called quoting to selectively suppress unwanted expressions.

• You can break a long command onto multiple lines by using '\ ' followed by enter.

↳ '\ ' (backslash) ignores the next character

↳ Ex:    ls -l \  
          -- reverse l  
          -- human-readable l  
          -- full time

## ★ Chapter 9: Permissions

### Commands:

chmod: modify file access rights

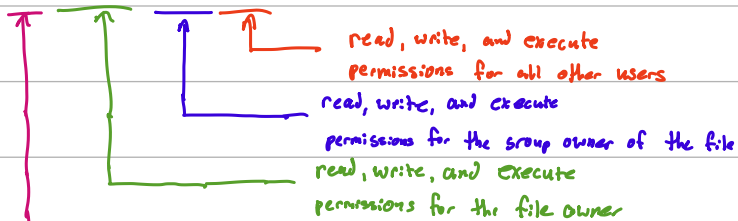
su: temporarily become the superuser

sudo: temporarily become the superuser

chown: change file ownership

chgrp: change a file's group ownership

• " - rwx rwx rwx "

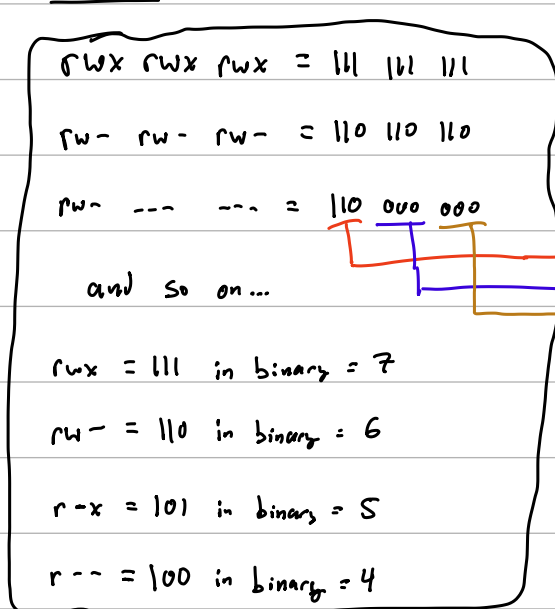


File type:

"-" indicates regular file

"d" indicates directory

## • chmod



chmod 600 some-file

↳ sets some-file to have read and write permissions for the owner, but the no one else has any permission.

## • Becoming superuser

• "sudo some\_command" → executes command as superuser

## • Changing Ownerships

• Files: sudo chown you some-file

• Groups: chgrp new\_group some-file

↳ new group ownership

## ★ Chapter 10: Job Control

### • Commands:

ps : list the progress running on the system

kill: send a signal to one or more processes

jobs: an alternative way of listing your own processes

bg: put a process in the background

fg: put a process in the foreground