Notes 9

Commands

grep

- Definition
    - Grep is used to search text inside a file. It works line by line, meaning it checks each line separately and prints only the lines that match the search criteria
- Formula
    - `grep + option + 'search crieria' + files`
        - search criteria - the word or pattern you looking for
        - file(s) one or more files to search
- Example
    - Search for any line that contains the word `"dracula"`
        - `grep 'dracula' ~/Documents/dracula.txt`
    - Search for dracula regardless of uppercase or lowercase letters
        - `grep -i 'dracula' ~/Documents/Books/dracula.txt`
    - Search for "dracula" ignoring case and show line numbers
        - `grep -in 'dracula' ~/Documents/Books/dracula.txt`
    - Show all line that do not contain the word "war"
        - `grep -v 'war' ~/Documents/Books/war-and-peace.txt`
- Display only the matched word "pride"
    - `grep -o 'pride' ~/Documents/Books/war-and-peace.txt`
- Show information about the current user
    - `grep -i $USER /etc/passwd`

Common option `grep` `-i` = ignore case (uppercase/lowercase does not matter) `-n` = show line numbers for matching lines

`-E` = Use extended regular expressions

`-G` = Use basic regular expressions

`-v` = Invert search (show lines that do not match)

`-o` = Show only the matched part of the line

`-c` = Count how many times a match occurs

`-w` = Match the whole word only

`r`, `-R` = Search recursively in directories.

---

awk

- Definition
  - `awk` is a text-processing scripting language. It reads input line by line and is mainly used to extract and print columns (fields) from files.
  - Can think of `awk` as "Take this file and show me specific columns"
- Formula:
  - `awk + options + {awk command} + file + file to save (optional)`
- Example
  - Print the first column of every line
    - `awk '{print $1}' ~/Documents/Csv/car.csv`
  - Print the first field of /etc/passwd
    - `awk -F:'{print $1}' /etc/passwd`
  - Print the last field
    - `awk -F: '{print $NF}' /etc/passwd`
  - Print first and last field
    - `awk -F: '{print $1, "=" , $NF}' /etc/passwd`

AWK variables

`$0` - Entire line `NR` - Line number `NF` - Number of Fields `FS` - Input field separator `OFS`- Output record separator `FILENAME` - Name of the file `IGNORECASE` - Ignore case

---

sed

- Definition
  - `sed` (stream editor) edits text without opening the file. It can search search, replace, insert, or delete text from files or standard output.
  - Can think of `sed` as edit text automatically.
- Formula
  - `sed options + sed script + file`

- Example
  - Replace "false" with "true" everywhere in a file
    - `sed 's/false/true/g' ~/Documents/sample_files/Json/joke.json`

Using the Pipe (`|`)

- Definition
  - The pipe symbol | is used to send the output of one command directly into another command as input. Instead of saving the output to a file first, the pipe lets commands work together in a chain.
  - Can think of it like, take the result of the command and pass it to the next one.
- Usage
  - `command1 | command2`
    - command 1 - produce output
    - command 2 - uses that output as input
- Examples
  - Search for the word "bash" inside the output of /etc/passwd
    - `cat /etc/passwd | grep "bash"`
  - Count how many lines contain the word "dracula"
    - `grep "dracula" ~/Documents/Books/dracula.txt | wc -1`
  - Display only file siezes and names from `ls` output
    - `ls -1h | awk '{print $5, $9}'`

Saving Command Output to a file (`>`)

- Definition
  - The `>` symbol redirects the output of a command into a file. *If the file exists, `>` will overwrite its contents*
- Formula
  - `command > filename`
- Example
  - Save a list of files into files.txt
    - `ls > files.txt`
  - Save search results into a files
    - `grep "error" log.txt > errors.txt`
  - Save the current date and time into a file
    - `date > today.txt`

Appending Command Output to a File (`>>`)

- ## Definition
  - The `>>` symbol adds outputs to the end of file without deleting existing content. Use this when you want to keep previous data and add more
  - Can think of it like, Add this output to the bottom of the file.
- ## Formula
  - `command >> filename`
- ## Example
  - Append a file list to file.txt
    - `ls >> files.txt`
  - Append matching lines to a log file
    - `grep "warning" log.txt >> warning.txt`
  - Append the current date to a history file
    - `date >> history.txt`