

# Final Assignment

---

## Question 1

---

### awk

- scripting language used for processing & displaying text. Performs operations line by line
- Formula:
  - `awk + options + {awk command} + file + file to save (optional)`
- EX: Print a list of all user in your system
  - `awk -F: '{print $1}' /etc/passwd` \*EX: extract first and third columns of a file
  - `awk '{ print $1, $3 }' file_name` \*EX: Display only the names of the cereals in the cereal.csv file.
  - `awk -F';' 'N > 3 {print $1}' cereal.csv`

### cat

- create, merge or print files in the standard output screen or to another file
- Formula:
  - `cat + option + Files to display`
- EX: Display the content of the `/etc/passwd` file.
  - `cat /etc/passwd`
- EX: Display the content of the `/etc/passwd` file with line numbers and the `$` to indicate the end of every line.
  - `cat -nE /etc/passwd`
- EX: Replace the `;` for a `,` in the `cereal.csv` file
  - `cat cereal.csv | tr ';' ','`

### cp

- Copies files and directories, including their content
- Formula:
  - `cp + files to copy + destination`
- EX: create directory `ws`, in `ws` directory create directory called `doc`
  - `mkdir ws/docs`
- EX: Download sample `doc`, `xls` & `pdf` from `ws`
  - `cp Downloads/sample`
  - `sample1 , sample1 , sample1`
- EX: Copy files to `docs` directory inside `ws`.
  - `cp Downloads/sample1.doc website/docs/`
  - `cp Downloads/sample1.pdf website/docs/`
  - `cp Downloads/sample1.xls website/docs`

### cut

- Extract a specific section of each line of a file

- Formula:
  - cut + option + Files
- EX: Display the first field of the /etc/passwd file.
  - cut -d ":" -f /etc/passwd
- EX: Display the first and last field of the /etc/passwd file.
  - cut -d ":" -f1,7 /etc/passwd
- EX: Display the first and last field of the /etc/passwd file with the = as the output delimiter.
  - cut -d ":" -f1,7 --output-delimiter="=" /etc/passwd
  - Display all the fields of the /etc/passwd file except the 3rd field.
    - cut -d ":" --complement -s -f3 /etc/passwd

## grep

- Grep allow you to find a word by searching the content of a file. This Linux command prints all lines containing the matching strings
- Formula:
  - grep + option + search criteria + file
- EX: Display every line that starts exactly with the word "love" in the book Dracula
  - grep '^love' dracula.txt
- EX: output only lines that have ip addr
  - my own ip addr
  - ip addr
  - ip addr | grep "inet"
- EX: how many users can log into my computer
  - grep -c "/bin/bash" /etc/passwd
- EX: Display every line that starts with an upper case letter or a number in the book Dracula
  - grep -E '^[A-Z]|[0-9]' dracula.txt

## head

- Display the top N number of lines of given file, by default prints first 10.
- Formula:
  - head + option + Files
- EX: Display the first 5 lines of a the /etc/passwd file.
  - head -5 /etc/passwd
- EX: Display the first 4 lines of multiple files using single command.
  - head example1.txt example2.txt
    - example1.txt
    - 1
    - 2
    - 3
    - 4
    - example2.txt
    - 1
    - 2
    - 3
    - 4

- EX: To redirect first 4 lines of example1.txt to a file named output1.txt.
  - `head example1.txt > output1.txt`
    - to check if successful run:
    - `cat output.txt`
    -

## ls

- will display the contents of the current directory. By default, ls lists files and directories in alphabetical order.
- Formula:
  - `ls file/directory`
- EX : List all log files located in lab6 directory.
  - `ls lab6/*.log`
- EX: List all the configuration files in the etc directory. (Configuration files have the extension of .conf)
  - `ls /etc/*.conf`
- EX: list the contents of the directory in a table format with columns included.
  - `ls -l`
- EX: List files and output the result to a file
  - `ls > output.txt`
- EX: List only the hidden files inside lab6 directory.
  - `ls .??*`
- EX: List all the files that start with a lowercase letter and have a number before the file extension.
  - `ls [a-z][0-9].*`

## man

- to display the user manual of any command that we can run on the terminal.
- Formula:
  - `man + command`
- EX: Display man page section of the passwd command
  - `man -f passwd`
- EX: Display man page for given word or regular expression.
  - `man -f file`

EX: The following allows users to look up the man pages and prints out short descriptions of the specified command in the terminal.

    - `man -f command name`
    - `man -f grep`

## mkdir

- allows the user to create directories also referred to as folders in some operating systems
- Formula:
  - `mkdir + name of directory`
- EX: In your home directory, create a directory called: lab5ChallengeQuestion and inside this new directory create one directory called assets you must achieve this in a single command.
- `mkdir -p lab5ChallengeQuestion/assets`

- ls lab5ChallengeQuestion/
  - assests
- EX: Change your current working directory to ~/lab5. Create 3 directories: pets,docs,songs. Inside each directory, create 2 subdirectories called: new and old.
  - cd lab/5
    - mkdir pets docs songs
    - mkdir pet/new pet/old docs/new docs/old songs/new songs/old
      - Display with tree
        - tree .
    - EX: create directory w/ parent(home) directory at the same time.
      - create directory: assets.image/small inside website directory
      - mkdir -p ~/website/assets/images/small

## mv

- EX: Create the directory at Home
  - mkdir ~/examples
    - cd ~
    - ls = example
- EX: Create the directory at desired location
  - mkdir /tmp/examples
    - cd /tmp
    - ls = examples
- EX: Download an image from this website: pexels.com & rename it newPicture.png. Move this image to your assets directory. You must move and rename the file in a single command.
  - mv Downloads/pexels-photo-454.jpeg lab5ChallengeQuestion/assets/newPicture.png
  - ls lab5ChallengeQuestion/assets
    - newPicture.png

## tac

Tac command is used for display content of file in reverse order.

- Formula:
  - tac + option + Files to display
- EX: Display the content of the /etc/passwd file in reverse order.
  - tac /etc/passwd
- EX: Display content of path using absolute path
  - tac ~/Documents/todo.md
- EX: The -b option attach separator before instead of after
- tac -b example1.txt example2.txt (two separate list)
  - list 3
  - list 2
  - list 1
  - list 3

- list 2
- list 1

## tail

- prints the last N number of data of the given input. By default, it prints the last 10 lines of the specified files.
- Formula:
  - tail + file
- EX: Display the last 5 lines of the /etc/passwd file.
  - tail -5 /etc/passwd
- EX: To limits output to the lines after line n or to the last n lines.
  - tail -n5 example1.txt
- EX: Display the account info stored in /etc/passwd of last user in your system.
  - tail -1 /etc/passwd

## touch

used to create a file without any content. The file created using the touch command is empty.

- Formula:
- touch file\_name
- EX: From the root of the file system, create a file in the lab5ChallengeQuestion directory called site.html
  - cd /
    - pwd
    - /
    - touch ~ lab5ChallengeQuestion/site.html
- EX: create several files.
  - touch example1.txt example2.csv
- EX: TO check if a file is created or not.Avoid creating file if already exist.
  - pwd
  - touch -c file\_name

## tr

- used for translating or deleting character from standard output
- Formula:
  - standard output | tr + set + set
- EX: Replace the ; for a , in the cereal.csv file.
  - cat cereal.csv | tr ';' ','
- EX: Delete specific characters use the -d option. This option deletes characters in the first set specified.
  - echo "Welcome To School" | tr -d W
- EX: convert lower case characters to upper case
  - cat school
    - WELCOME TO high school

- `cat school | tr [a-z] [A-Z]`
  - WELCOME TO HIGH SCHOOL

## tree

- Displays directory structure of a file system in a tree-like format. It shows nested structure of directories, sub-directories, and files in a hierarchical tree-like structure.
- Formula:
  - `tree + directory`
- EX: display only directories in directory structure.
  - `tree -d`
- EX: display tree of directory
  - `tree wallpapers/`
- EX: Display hidden files and directories
  - `tree -a`

## Question 2

---

- How to work with multiple terminals open?
  - Once a terminal window is open on the top left side of the terminal window a small square box appear. Clicking once an additional terminal window appear or as many as needed with multiple click. By right clicking on top of terminal window on the black bar near user name several option appear to detach terminal or move terminal.
- How to work with manual pages?
  - To run the man command:
    - `man + command line`
- It provides a detailed view of the command which such as name, synopsis, description
  - To exit man page press letter q or Control C
- How to parse (search) for specific words in the manual page
  - `man -k 'name'`
- How to redirect output (> and |)
  - Overwrite redirection is useful when you want to store/save the output of a command to a file and replace all the existing content of that file.
  - For example, if you run a command that gives a report, and you want to save the report to the existing file of the previous report you can use overwrite redirection.
    - `cat source.txt > destination.txt`
- The | command is called pipe. It is used to pipe, or transfer, the standard output from the command on its left into the standard input of the command on its right.

- file name file1.txt and want to redirect it to a file name 'example.txt'.
  - `ls | grep 'file' > example.txt`
- How to append the output of a command to a file
  - To append is simply add the data to the file without erasing existing data.
  - `cat source.txt >> destination.txt`
- How to use wildcards
  - For copying and moving multiple files at the same time
  - The main wildcard is a star or asterisk (\*). Star alone matches anything & nothing & matches any number of character.
  - EX: list all files that have any letter before string "file" & after as well
    - `(*asterisk) file. (*asterisk)`
  - The ? wildcard metacharacter matches precisely one character. Proves useful with hidden files(dot files)
  - EX: List all hidden files in parent directory
    - `ls ../.??*`
  - EX: List all files that have two character b/n letter b & k
    - `ls b??k*`
- How to use brace expansion
  - For creating entire directory structures in a single command
  - Create whole directory structure in single command:
  - `mkdir -p music/{jazz,rock}/{mp3files,videos,newfiles}/new{1..3}`