Final Exam Study Assignment

Question 1

awk

- Description:
  - Awk is a scripting language used for processing and displaying text. Awk can work with a text file or from standard output. Awk performs operations line by line.
- Formula/Syntax:
  - `awk + options + {awk command} + file + file to save (optional)`
- Examples:
  - Print the first column of every line of a file
    - `awk '{print $1}' ~/Documents/Csv/cars.cvs`
  - Print the first field of /etc/passwd file
    - `awk -F: '{print $1}' /etc/passwd`
  - Convert the first field to upper/lower case
    - `awk -F: '{print toupper($1)}'`

cat

- Description:
  - The cat command is used for displaying the content of a file. Also used for concatenating files.
- Formula/Syntax:
  - `cat + option + file(s) to display`
- Examples:
  - Display the content of a file located in the pwd
    - `cat todo.lst`
  - Display the content of a file with line numbers
    - `cat -n ~/Documents/todo.md`
  - Display the content of a file with line numbers excluding empty line
    - `cat -b ~/Documents/todo.md`

cp

- Description:
  - The cp command copies files or directories from a source to a destination.
- Formula/Syntax:
  - `cp + files to copy + destination`
- Examples:
  - To copy a file
    - `cp Downloads/wallpapers.zip Pictures/`
  - Copy the content of a directory to another directory
    - `cp Downloads/wallpapers.zip Pictures/`
  - To copy multiple files in a single command

        - `sudo cp -r script.sh program.py home.html assets/ /var/www/html/`

grep

- Description:
    - Grep is used to search text in given file. Grep works line by line basis (it matches the search criteria ina line by line basis).
- Formula/Syntax:
    - `grep + option + search criteria + file(s)`
- Examples:
    - Search any line that contains the word "dracula" in the given file
        - `grep 'dracula' ~/Documents/dracula.txt`
    - Search for a given string inside files in a given directory
        - `grep -iR 'conf' /etc/`
    - Search for all the lines that end with the string "nologin"
        - `grep -n 'nologin$' /etc/passwd`

ls

- Description:
    - The ls command is used to list files inside a given directory. The ls command can be used without arguments.
- Formula/Syntax:
    - `ls + option + directory to list`
- Examples:
    - List all the files inside the current working directory including hidden files
        - `ls -a`
    - Long list all the files inside a given directory recursively
        - `ls -lR ~/Pictures`
    - List all the options of the ls command
        - `ls --help`

man

- Description:
    - The man command is used to show pages of documentation files that describe Linux shell commands, executable programs, system calls, special files, and so forth. Man pages are not step by step guides, but instead quick references.
- Formula/Syntax:
    - `man + command`
- Examples:
    - Open the man page of the passwd command
        - `man passwd`
    - Show the man page section of the passwd command
        - `man -f passwd`
    - Searches for a man page for a given word or regular expression or phrase
        - `man -k file`

mkdir

- Description:
  - The mkdir command is used to make directories.
- Formula/Syntax:
  - `mkdir + the name of the directory`
- Examples:
  - Create a directory in the present working directory
    - `mkdir wallpapers`
  - Create a directory with a space in the name
    - `mkdir wallpapers/new\ cars`
    - `mkdir wallpapers/'cities usa'`
  - Create multiple directories
    - `mkdir wallpapers/cars wallpapers/cities wallpapers/forest`

tac

- Description:
  - The command is used for displaying the content of a file in reverse order. Just like cat, tac concatenates files and displays the output of the concatenation.
- Formula/Syntax:
  - `tac + option + file(s) to display`
- Examples:
  - Display the content of a file located in the pwd
    - `tac todo.md`
  - Display the content of file using absolute path
    - `tac ~/Documents/todo.md`
  - Display the content of a file with line numbers excluding empty line
    - `tac -b ~/Documents/todo.md`

tail

- Description:
  - The tail command displays the last N number of lines of a given file. By default, it prints the last 10lines. If more than one file name is provided then data from each file is preceded by its file name.
- Formula/Syntax:
  - `tail + option + file`
- Examples:
  - Display the last 10 lines of a file
    - `tail ~/Documents/Book/dracula.txt`
  - Display the last 5 lines of a file
    - `tail -5 ~/Documents/Book/dracula.txt`
  - Display the account information stored in /etc/passwd of the first user in your system
    - `tail -1 /etc/passwd`

touch

- Description:
  - The touch command is used to create files.
- Formula/Syntax:
  - `touch + name of file`
- Examples:
  - Create a file called foods
    - `touch foods`
  - Create several files
    - `touch list_of_foods.txt recipes.docx cooks.csv`
  - Create a file with a space in its name
    - `touch "list of foods.txt"`

tr

- Description:
  - The tr command is used for translating or deleting characters from standard output.
- Formula/Syntax:
  - `standard output | tr + option + set + set`
- Examples:
  - Translate one character to another (For example a period with a comma)
    - `cat file.txt | tr '.' ','`
  - Translate white space into tabs
    - `cat program.py | tr "[:space:]" '/t'`
  - Translate tabs into space
    - `cat file.py | tr -s "[:space:]" ' '`

tree

- Description:
  - The tree command will display all files and directories in a given directory in a tree like format. It takes a relative path, absolute path, or no argument.
- Formula/Syntax:
  - `tree + options + directory to list`
- Examples:
  - Display the directory tree of the current directory
    - `tree Documents`
  - Display the directory tree with specific depth
    - `tree -L 3 Downloads`
  - Display only directories and exclude hidden files
    - `tree -d -I '.*'`

Question 2

- **How to work with multiple terminals open?**
  - You work with multiple directories by having your main work and commands on 1 terminal and using commands like `ls`, `tree`, and `man` on the other terminal for references and guides.

- **How to work with manual pages?**
  - To work with manual pages you would use the `man` command with a directory or command to display information about the command or directory.
- **How to parse (search) for specific words in the manual page**
  - You would use `/` to find specific words in the manual page.
- **How to redirect output (> and |)?**
  - You would use `>` command to save the output of a command to a file. If the `>` command is used on an existing file that contains data then that data will be overwritten. You would use `|` to pipe the output of one command as the input to another command. This allows you to chain commands together.
- **How to append the output of a command to a file?**
  - You would use `>>` command to append the output of a command to a file.
- **How to use wildcards (For copying and moving multiple files at the same time)?**
  - `cp + file + source | mv + file + destination`
- **How to use brace expansion (For creating entire directory structures in a single command)?**
  - `mkdir -p + directory/{directories or files}/{directories or files}/{directories or files}`
  - `mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}/new{1..3}`