Final Exam Assignment

Question 1

awk

- Description:
  - Awk is a scripting language that is used for processing and displaying text.
- Syntax/Formula:
  - `awk + options + {awk command} + file + file to save (optional)`
- Examples:
  - Print the first column of every line in a file
    - `awk '{print $1}' /etc/passwd`
  - Print first and last field of a file
    - `awk -F: '{print $1," = ",$NF}' /etc/passwd`
  - Print first and 4th field with different field separator
    - `awk -F: '{OFS="="}{print $1,$4}' /etc/passwd`
  - Print a file from a given line (exclude first 2 lines)
    - `awk 'NR > 3 { print }' /etc/passwd`

cat

- Description:
  - The cat command is used for displaying the content of a file.
- Syntax/Formula:
  - `cat + option + file(s) to display`
- Examples:
  - Display the content of a file in pwd
    - `cat finalexam.lst`
  - Display the content of a file with line numbers
    - `cat -n ~/Documents/finalexam.md`
  - Display the content of a file with a $ at the end of every line
    - `cat -E ~/Documents/finalexam.md`

cp

- Description:
  - The cp command copies files or directories from a source to a destination.
- Syntax/Formula:
  - `cp + files to copy + destination`
- Example:
  - Copy a file
    - `cp Downloads/finalexam.txt Documents/`
  - Copy the content of a directory to another directory
    - `cp Downloads/wallpapers/* ~/Pictures/`
  - Copy multiple files in a single command

    - `sudo cp -r program.py finalexam.md mywebsite.html assets/ /var/www/html/`

cut

- Description:
    - The cut command is used to extract a specific section of each line and display it to the screen.
- Syntax/Formula:
    - `cut + option + file(s)`
- Examples:
    - Cut a file using a delimiter but change the delimiter in the output
        - `cut -d ':' -f1,7 --output-delimiter=' => ' /etc/passwd`
    - Cut a file while excluding a given field
        - `cut -d ',' --compliment -s -f3 users.txt`
    - Display a list of all the users in your system with their login shell
        - `cut -d ':' -f1,7 /etc/passwd`

grep

- Description:
    - The grep command is used to search text in a given file in a line by line basis.
- Syntax/Formula:
    - `grep + option + search criteria + file(s)`
- Example:
    - Search any line that contains the word 'dracula regardless of the case
        - `grep -i 'dracula' ~/Documents/Books/dracula.txt`
    - Search any line that contains the word 'dracula regardless fo the case with line numbers
        - `grep -in 'dracula' ~/Documents/Books/dracula.txt`
    - Search for all lines that do not contain the word 'war
        - `grep -v 'war ~/Documents/Books/war-and-peace.txt`
    - Search and match only the word
        - `grep -o 'love' ~/Documents/Books/bible.txt`

head

- Description:
    - The head command displays the top N number of lines in a given file.
- Syntax/Formula:
    - `head + option + file(s)`
- Examples:
    - Display the first 10 lines of a file
        - `head ~/Documents/Books/bible.txt`
    - Display the first 5 lines of a file
        - `head -5 ~/Documents/Books/bible.txt`
    - Display the first 3 lines of a file
        - `head -3 ~Documents/Books/bible.txt`

ls

- Description:
  - The ls command is used to list the content of a given directory or the file/directory itself.
- Syntax/Formula:
  - `ls + option + directory to list`
- Examples:
  - List the content of the present working directory (pwd)
    - `ls`
  - List all the files in a given directory sorted by file size
    - `ls -S ~/Documents`
  - List all the files in a given directory sorted by extension
    - `ls -X ~/Documents`
  - List all the files the options od the ls command
    - `ls --help`

man

- Description:
  - The man command is referred to as a manual page that describe Linux shell commands.
- Syntax/Formula:
  - `man + command`
- Examples:
  - Open the man page of the ls command
    - `man ls`
  - Open the page man page of the cut command
    - `man cut`
  - Open the man page for the awk command
    - `man awk`

mkdir

- Description:
  - The mkdir command is used to create either a single directory or multiple directories.
- Syntax/Formula:
  - `mkdir + the name of the directory`
- Examples:
  - Create a directory in the present working directory
    - `mkdir wallpapers`
  - Create a directory in a different directory using relative path
    - `mkdir wallpapers/ocean`
  - Create a directory in a different directory using absolute path
    - `mkdir ~/wallpapers/cave`
  - Create multiple directories
    - `mkdir wallpapers/cars wallpapers/sports wallpapers/dark`

mv

- Description:
  - The mv command moves and can also rename files.

- Syntax/Formula:
  - `mv + source + destination`
- Examples:
  - To move a file from one directory to another using relative path
    - `mv Downloads/sports.pdf Documents/`
  - To move multiple directories or files to a different directory
    - `mv games/ wallpapers/ music/ /media/student/flashdrive/`
  - To rename a file
    - `mv homework.docx nohomework.docx`
  - To move and rename a file in the same command
    - `mv Downloads/nohomework.docx Documents/lotsofhomework.docx`

tac

- Description:
  - The tac command is used to display content in reverse order.
- Syntax/Formula:
  - `tac + option + file(s)to display`
- Examples:
  - Display the content of a file in pwd
    - `tac dog.md`
  - Display the content of a file using absolute path
    - `tac ~/Documents/dogfood.txt`
  - Display the content of a file using relative path
    - `tac Documents/sports.md`

tail

- Description:
  - The tail command displays the last N number of lines of a given file.
- Syntax/Formula:
  - `tail + option + file`
- Example:
  - Display the last 10 lines of a file
    - `tail ~/Documents/Books/bible.txt`
  - Display the last 5 lines of a file
    - `tail -5 ~/Documents/Books/bible.txt`
  - Display the last 3 lines of a file
    - `tail -3 ~/Documents/Books/bible.txt`

touch

- Description:
  - The touch command is used for creating files.
- Syntax/Formula:
  - `touch + file(s)`
- Examples:
  - To create a file called list

        - `touch list`
    - To create several files
        - `touch list_of_games.txt script.py names.csv`
    - To create a file using absolute path
        - `touch ~/Downloads/games.txt`

tr

- Description:
    - The tr command is used for translating or deleting characters from standard output.
- Syntax/Formula:
    - `Standard output | tr + option + set + set`
- Examples:
    - Translate one character to another (A period with a comma for example)
        - `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 is a recursive directory listing program that produces a depth-indented listing of files.
- Syntax/Formula:
    - `tree + option + directory`
- Examples:
    - List all the files and directories in the current working directory
        - `tree`
    - List all of the files and directories including hidden files
        - `tree -a ~/Pictures`
    - List only the directories
        - `tree -d ~/Documents`

Question 2

How to work with multiple terminals open?

- To work with multiple terminals open, you can use Tilix and split the terminal there or you can open many terminal applications. This is convenient as it gives you the space to test out commands and to have the help page ready to find an option.

How to work with manual pages?

- Working with manual pages is very simple. If you do not know what a command is or any of the options the command has, you can refer to the manual page as a guide and a solution to find your answer.

How to parse (search) for specific words in the manual page

- To search for specific words in the manual page, you can combine the man command with the grep command by using pipe.
- Example:
    - `man ls | grep "sort"`

How to redirect output (> and |)

- To redirect output with ">" to append output or add more, you use this formula: `Command output + > + file`.
- Example:
    - Save the output of a command to a file
        - `ls -lA ~ > files.txt`
- To redirect output using | (Pipe command) you use this formula: `command_1 | command_2 | command_3 | ...`
- Example:
    - Display only the 2nd line in a file
        - `head -2 file.lst | tail -1`

How to append the output of a command to a file

- To append the output, you add more to a file instead of overwriting. To do this I will show an example: `ls -la > allmyfiles.lst` **How to use wildcards**
- The * wildcard is a star that matches anything and nothing and can match numbers as well as characters. For example: `ls *.txt` will match all the files that end in .txt.
- The ? wildcard matches precisely one character. For example, to list all hidden files use: `ls .??*` which matches all files that start with a . or .. and have any character after it.
- The [] wildcard or brackets wildcard matches a single character in a range. For example, to match all files that have a vowel after letter f you use: `ls f[aeiou]*`.

How to use brace expansion

- Brace expansion "{}" allows you to generate arbitrary strings. For example, you can do: `mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}/new{1..3}`