Final Exam Study

Question 1

Awk

- Description:
  - Is a scripting language used for processing and displaying text.
- 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.csv`
  - print the last field of a file: `awk -F: "{print $NF}" /etc/passwd`
  - Print a file from a given line: `awk "NR > 3 { print }" /etc/passwd`

Cat

- Description:
  - Is used for displaying the content of a file.
- Formula/syntax:
  - `cat + option + file(s) to display`
- Examples:
  - Display the content of file in the pwd: `cat todo.lst`
  - Display the content with absolute path: `cat ~/Documents/todo.lst`
  - Display the content with line numbers excluding empty lines: `cat -b ~/Documents/todo.md`

cp

- Description:
  - Copies files/directories from a source to a destination.
- Formula/syntax:
  - `cp + files to copy + destination`
  - `cp -r + directory to copy + destination` (directories)
- Examples:
  - To copy a file: `cp Downloads/wallpapers.zip Pictures/`
  - To copy a directory (absolute path): `cp -r ~/Downloads/wallpapers ~/Pictures/`
  - To copy the content of a directory to another directory: `cp Downloads/wallpapers/* ~/Pictures/`

cut

- Description:
  - Is used to extract a specific section of each line of a file and display it to the screen.

- Formula/syntax:
    - `cut + option + file(s)`
- Examples:
    - Display a list of all the users in your system: `cut -d ':' -f1 etc/passwd`
    - Display all the users in your system with their login shell: `cut -d ';' -f1,7 /etc/passwd`
    - cut a file using a delimiter but changing the delimiter in the output: `cut -d ':' -f1,7 --output-delimiter=' => ' /etc/passwd`
        - Command output:

```
usbmux => /usr/sbin/nologin
dnsmasq => /usr/sbin/nologin
kernoops => /usr/sbin/nologin
avahi => /usr/sbin/nologin
cups-pk-helper => /usr/sbin/nologin
rtkit => /usr/sbin/nologin
whoopsie => /bin/false
sssd => /usr/sbin/nologin
speech-dispatcher => /bin/false
fwupd-refresh => /usr/sbin/nologin
nm-openvpn => /usr/sbin/nologin
saned => /usr/sbin/nologin
colord => /usr/sbin/nologin
geoclue => /usr/sbin/nologin
pulse => /usr/sbin/nologin
gnome-initial-setup => /bin/false
hplip => /bin/false
gdm => /bin/false
ajtavarez => /bin/bash
vboxadd => /bin/false
_flatpak => /usr/sbin/nologin
```

Grep

- Description:
    - Is used to search text in given file. Grep works line by line basis.
- Formula/syntax:
    - `grep + option + search criteria + file(s)`
- Examples
    - Search any line that contains the word "Dracula" in the given line: `grep 'Dracula' ~/Documents/dracula.txt`
    - Search with no case sensibility: `grep -i 'Dracula' ~/Documents/dracula.txt`
    - Search for all the lines that do not contain the word 'war': `grep -v 'war' ~/Documents/Books/war-and-peace.tx`

Head

- Description:
    - Displays the top N number of lines of a given file.

- Formula/syntax:
  - `Head + option + file(s)`
- Examples:
  - Display the first 10 lines of a file: `head ~/Documents/Book/dracula.txt`
  - Display the first 5 lines of a file: `head -5 ~/Documents/Book/dracula.txt`
  - Display the account information stored of the first user in your system: `head -1 /etc/passwd/`

Ls

- Description:
  - Used for displaying all the files inside of a given directory.
- Formula/syntax:
  - `ls + file`
- Examples:
  - List the content of the present working directory: `ls`
  - list all the files including hidden files: `ls -a`
  - List all the files in a given directory sorted by file extension: `ls -X ~/Documents`

Man

- Description:
  - (manual) are documentation files that describe Linux shell commands, executable programs, systems calls, etc.
- Formula/syntax:
  - `man + command`
- Examples:
  - To view the manual of 'ls' command: `man ls`
  - Open the man page of the passwd command: `man passwd`
  - Show all the available pages of a command: `man -a passwd`

Mkdir

- Description:
  - Is used for creating a single directory or multiple directories.
- Formula/syntax:
  - `mkdir + the name of the directory`
- Examples:
  - Create a directory with relative path: `mkdir Wallpapers/ocean`
  - Create multiple directories: `mkdir wallpapers/cars wallpapers/cities wallpapers/forest`
  - Create a directory with a parent directory at the same time: `mkdir -p wallpapers_others/movies`

Mv

- Description:
  - Moves and renames directories.

- Formula/syntax:
  - `mv + source + destination`
- Examples:
  - For renaming files/directories the formula remains the same:`mv + file/directory to rename + new name`
  - To move a file from a directory to another with relative path: `sudo mv ~/Downloads/theme /usr/share/themes`
  - To rename and move a file: `mv Downloads/cis106homework.docx Documents/new_cis106homework.docx`

Tac

- Description:
  - Is used for displaying the content of a file in reverse order.
- Formula/syntax:
  - `tac + option + file(s) to display`
- Examples:
  - Display the content of a file in the pwd:`tac todo.md`
  - Display the content of a file with absolute path: `tac ~/Documents/todo.md`

Tail

- Description:
  - Displays the last N number of lines of a given file.
- 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`

Touch

- Description:
  - Is used for creating files.
- Formula/syntax:
  - `touch + name of the file`
- Examples:
  - To create a file: `touch list`
  - To create several files: `touch list.txt script.py names.csv`
  - To create with relative path: `touch Downloads/games2.txt`

Tr

- Description:
  - The tr command in Ubuntu is a text-processing utility that is used to translate, delete, or squeeze characters.
- Formula/syntax:
  - `standard output | tr + [options] + SET1 + SET2`

- Examples:
  - To replace a character `echo "HELLO" | tr 'el' 'EL'`
  - Uppercase the lowercase and vice versa: `cat file.txt | tr '[:lower:]' '[:upper:]'`

Tree

- Description;
  - Is a useful utility that displays the directory structure in a tree-like format
- Formula/syntax:
  - `tree + options + directory`
- Examples:
  - Display tree the current directory:`tree`
  - Display with file details: `tree -l`
  - Displays with hidden files: `tree -a`

Question 2

- How to work with multiple terminals open?

  - open a terminal and then open another terminal and set them side by side. One option is tillix and split the terminal as needed.

- How to work with manual pages?

  - To navigate the man page of a command, you can use the arrow key or the man command internal shortcuts.

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

  - Using the command `man -k file`.

- How to redirect output (> and |) and append the output of a command to a file?

  - Append '>' means to add more to a file instead of overwriting its content.
    - To save the output of a command to a file: `ls -la ~ > all-files-in-home.txt`
  - The pipe '|' allows you to redirect the standard output of a command.
    - usage: `command_1 | command_2 | command_3 | ..... | command_N`

- How to use wildcards?

  - This command represent letters and characters used to specify a file name for searches.
    - `ls *.txt` will match all files that end in .txt.
    - `ls .??*` will match all files that start with a '.' or '..' .
    - `ls f[aeiou]*` will match all files that have a vowel after letter f.

- For copying and moving multiple files at the same time:

  - To copy multiple files: `sudo cp -r script.sh program.py home.html assets/ /var/www/html/`

  - For moving multiple files/directories: `mv games/ wallpapers/ rockmusic/ /media/student/flashdrive`

- How to use brace expansion:

  - '{}' Is a feature in the shell that allows you to generate multiple strings or sequences based on a pattern.

- For creating entire directory structures in a single command.

  - `mkdir -p files_organized/{audio/{aac,mp3},imgs/{gifts,jpgs,pngs},progs/{python/go/ruby/bahs}}`