final assignment

question 1

awk

- description:
  - awk is a scripting language used for processing and displaying text.
- formula:
- `awk + options + {awk command} + file + file to save(optional)`
- examples:
  - print the last field of a file
    - `awk '{print $NF}' ~/Documents/Books/bible.txt`
  - print the 2nd, 5th, and last line of a file
    - `awk '{print $2,$5,$NF}' ~/Documents/Books/dracula.txt`
  - convert the first field to upper/lower case
    - `awk '{print toupper($1)}' ~/Documents/Books/dracula.txt`

cat

- description:
  - the cat command is used for displaying the content of a file
- formula:
- `cat + option + file(s) to display`
- examples:
  - display the content of a file using absolute path
    - `cat ~/Documents/Books/pride-and-prejudice`
  - display the content of a file with line numbers
    - `cat -n ~/Documents/Books/bible.txt`
  - display the content of a file with a $ at the end of every line
    - `cat -E ~/Documents/Books/dracula.txt`

cp

- description:
  - the cp command copies files/directories from a source to a destination
- formula:
- `cp + files to copy + destination`(files) `cp -r + directory to copy + destination`(directories)
- examples:
  - to copy a file
    - `cp Downloads/wallpapers.zip Pictures/`
  - to copy a directory with absolute path
    - `cp -r ~/Downloads/wallpapers ~/Pictures/`
  - to copy multiple files in a single command

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

cut

- description:
    - the cut command is used to extract a specific section of each line of a file and display it on screen
- formula:
- `cut + option + file(s)`
- examples:
    - display a list of all the users in your system
        - `cut -d ':' -f1 /etc/passwd`
    - cut a file using a delimiter but changing the delimiter in the output
        - `cut -d ':' -f1,7 --output-delimiter=' => ' /etc/passwd`
    - display a list of all the users in your system with their login shell
        - `cut -d ':' -f1,7 /etc/passwd`

grep

- description:
    - grep is used to search text in a given file, works on a line by line basis
- formula:
- `grep + option + search criteria + file(s)`
- examples:
    - search the word jesus regardless of case
        - `grep -i 'jesus' ~/Documents/Books/bible.txt`
    - search for a word with line number for every line matched
        - `grep -in 'jesus ~/Documents/Books/bible.txt`
    - display the exact matched string with line number
        - `grep -on 'God' ~/Documents/Books/bible.txt`

head

- description:
    - the head command displays the top N number of lines of a given file
- formula:
- `head + option + file(s)`
- examples:
    - display the first 7 lines of a file
        - `head -7 ~/Documents/Books/war-and-peace.txt`
    - display the first 10 lines of a file
        - `head ~/Documents/Books/pride-and-prejudice.txt`
    - display the first line of a file
        - `head -1 ~/Documents/Books/bible.txt`

ls

- description:

    - used for displaying all the files inside a given directory
- formula:
- `ls + option + directory to list`
- examples:
    - list all the files in a given directory
        - `ls -a ~/Documents/Books`
    - list all the files in a given directory by file size
        - `ls -S ~/Documents/Books`
    - list all the options of the ls command
        - `ls --help`

man

- description:
    - man (manual) pages are documentation files that describe linux shell
- formula:
- `man + command`
- examples:
    - ls manual
        - `man ls`
    - grep manual
        - `man grep`
    - mkdir manual
        - `man mkdir`

mkdir

- description:
    - mkdir is used for creating a single directory or multiple directories
- formula:
- `mkdir + name of the directory`
- examples:
    - create a directory in a different directory using absolute path
        - `mkdir games/pokemon`
    - create a directory with a parent directory at the same time
        - `mkdir -p games/pokemon games/castlevania games/godofwar`
    - create a directory in the present working directory
        - `mkdir games`

mv

- description:
    - mv command moves and renames directories
- formula:
- `mv + source + destination`(for moving) `mv + file/directory to rename + new name`(for renaming)
- examples:
    - to rename a file

        - `mv castlevania.txt dawnofsorrow.txt`
    - to move and rename a file at the same time
        - `mv Downloads/pokemon.txt Documents/alphasapphire.txt`
    - to move a file from one directory to another using absolute path
        - `mv ~/Documents/hiphop.txt ~/Music`

tac

- description:
    - the tac command is used for displaying the content of a file in reverse order
- formula:
- `tac + option + file(s) to display`
- examples:
    - display the content of a file using absolute path
        - `tac ~Documents/Csv/cars.csv`
    - display the content of a file with line numbers
        - `tac -n ~/Documents/Books/bible.txt`
    - display the content of a file suppressing repeating empty lines to a single empty line
        - `cat -s ~/Documents/Books/war-and-peace.txt`

tail

- description:
    - the tail command displays the last N lines of a given file
- formula:
- `tail + option + file(s)`
- examples:
    - display the last 10 lines of a file
        - `tail ~/Documents/Books/dracula.txt`
    - display the last 5 lines of a file
        - `tail -5 ~/Documents/Books/dracula.txt`
    - display the last line of a file
        - `tail -1 ~/Documents/Books/war-and-peace.txt`

touch

- description:
    - touch is used for creating files
- formula:
- `touch + file name`
- examples:
    - create a text file called hip hop
        - `touch hiphiop.txt`
    - create several files at once
        - `touch hiphop.txt python.py education.csv`
    - create a file using absolute path
        - `touch ~/Documents/jujutsukaisen.txt`

tr

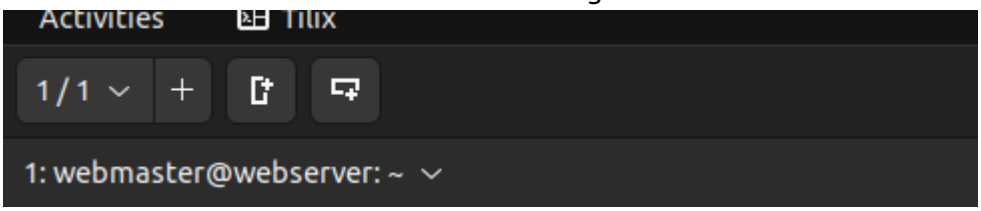
- description:
  - the tr command is used for translating or deleting characters from standard output
- formula:
- `standard output | tr + option + set + set`
- examples:
  - translate a period to a comma
    - `cat dracula.txt | tr '.' ','`
  - translate white space into tabs
    - `cat war-and-peace.txt | tr "[:space:]" '\t'`
  - translate tabs into space
    - `cat bible.txt | tr -s "[:space:]" ' '`

tree

- description:
  - the tree is a recursive directory listing program that produces a depth-indented listing of files. With no arguments, the tree lists the files in the current directory. When directory arguments are given, the tree lists all the files or directories found in the given directories each in turn.
- formula:
- `tree + options`
- examples:
  - display tree of Books directory using absolute path
    - `tree ~/Documents/Books`
  - display tree of whole system
    - `tree`
  - Prints the output by last modification time instead of alphabetically.
    - `tree -t ~/Music`

How to work with multiple terminals open?

to work with multiple terminals you can click on the add terminal symbol at the top left of the terminal you can insert the terminal underneath or to the right of the terminal screen



How to work with manual pages?

to work with the manual you type man followed by the command you want the manual of for example: `man mkdir`

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

to search for a specific word in the manual page we use the | symbol alongside grep and `man ls | grep "human-readable"`

How to redirect output (> and |)

to save standard output the > symbol is used, the formula for this is `command output + > + file`. if i want to redirect the standard output of a command then the | symbol is used for example : `command 1 | command 2 |command N`

How to append the output of a command to a file

to append the output of a file the >> symbol is used this will add whatever command i execute to the end of the file for example: `ls -la >> allmyfiles.lst`

How to use wildcards for copying and moving multiple files at the same time

wildcards makes it easy to move multiple files or copy multiple files for example if i wanted to move all my files in my present directory that end with a 3 letter extension but i dont care what comes before it or what the extension is then i would type `mv *.??? ~/Music` if i wanted to copy the exact file types from the new directory to my current then i would type `cp ~/Music/*.??? ~/Documents`

How to use brace expansion for creating entire directory structures in a single command

to use brace expansion for multiple directories you use the command `mkdir -p games/{action,rpg}/{level,items,loot}/savefile{1..3}` then you can use `tree` to check your directory structure