Week Report 7

Cat command

- The cat command is used to display the content of a file.
- Formula:
  - `cat + option + file(s) to display`
- Examples:
  - 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 lines:
    - `cat -b ~/Documents/todo.md`

Tac command

- 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 in the pwd:
    - `tac todo.md`
  - Display the content of a file using absolute path:
    - `tac ~/Documents/todo.md`

Head command

- The head command displays the top N number of lines of a given file. By default, it prints the first 10 lines. If more than one file name is provided, then the data from each file is preceded by its file name.
- Formula:
  - `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`

Tail command

- The tail command displays the last N number of lines of a given file. By default, it prints the last 10 lines. If more than one file name is provided, then data from each file is preceded by its file name.
- Formula:
  - `tail + option +file(s)`
- 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`

Cut command

- The cut command is used to extract a specific section of each line of a file and display it to the screen.
- Formula:
  - `cut + option + file(s)`
- Examples:
  - Display a list of all the users in your system:
    - `cut -d ':' -f1 /etc/passwd`
  - Display a list of all the users in your system with their login shell:
    - `cut -d ':' -f1,7 /etc/passwd`

Paste command

- The paste command is used for joining files horizontally in columns.
- Formula:
  - `paste + option + files`
- Examples:
  - Merge two files:
    - `paste users.lst ip_address.lst`
  - Merge two files using a different delimiter:
    - `paste -d ":" users1.lst ip_addresses.lst`

Sort command

- The sort command is used for sorting files in alphabetical, reverse, by number, or by month order.
- Formula:
  - `sort + option + file`
- Examples:
  - Sort a file and save the output to a new file:
    - `sort -o sorted.lst users.lst`
  - Sort a file in reverse order:
    - `sort -r users.txt`

Wc command

- The wc command is used for printing the number of lines, characters, and bytes in a file.
- Formula:
  - `wc + option + file(s)`
- Examples:
  - Display the number of characters in a file:
    - `wc -m users.txt`
  - Display the number of words in a file:
    - `wc -w users.txt`

Tr command

- The tr command is used for translating or deleting characters from standard output.
- Formula:
  - `Standard output | tr + option + set + set`
- Examples:
  - Translate white space into tabs:
    - `cat program.py | tr "[:space:]" '\t'`
  - Translate tabs into space:
    - `cat file.py | tr -s "[:space:]" ' '`

Diff command

- The diff command compares files and displays the differences between them.
- Formula:
  - `diff + option + file1 + file2`
- Examples:
  - Display the difference between two files:
    - `diff cars.csv cars-backup.csv`
  - Display the difference between two files in a column format:
    - `diff -y cars.csv cars-backup.csv`

Grep command

- Grep is used to search text in given file. Grep works line by line basis meaning it matches the search criteria in a line by line basis.
- Formula:
  - `grep + option + search criteria + file(s)`
- Examples:
  - Search any line that contains the word 'dracula' regardless of the case:
    - `grep -i 'dracula' ~/Documents/Books/dracula.txt`
  - Search for all the lines that do not contain the word 'war':
    - `grep -v 'war' ~/Documents/Books/war-and-peace.txt`

Awk command

- Awk is a scripting language used for processing and displaying text. Awk performs operations line by line.
- Formula:
  - `awk + options + {awk command} + file + file to save (optional)`
- Examples:
  - Print first field of /etc/passwd file:
    - `awk -F: '{print$1}' /etc/passwd`
  - Print the last field of the /etc/passwd file:
    - `awk -F: '{print $NF}' /etc/passwd`
  - Print the first and last field of the /etc/passwd file:
    - `awk -F: '{print $1," = ",$NF}' /etc/passwd`
  - Print the first and third field with line numbers:
    - `awk -F: '{print NR,$1,$3}' /etc/passwd`
  - Start printing a file from a given line (exclude the first 2 lines):

        - `awk 'NR > 3 { print }' /etc/passwd`

Sed command

- SED is a stream editor that perform operations on files and standard output. For instance, it can search, find and replace, insert, and delete. By using SED, you can edit files without opening them.
- Formula:
    - `sed options + sed script + file`
- Example:
    - Replacing a string in a given file (replace pizza for rice):
        - `sed 's/pizza/rice/' shopping-list.lst`
    - Replacing the number of occurrences of a pattern in a file:
        - `sed 's/pizza/rice/4' shopping-list.lst`
    - Replacing all the occurrences of the pattern in a file:
        - `sed 's/pizza/rice/g' shopping-list.lst`
    - Replacing from the given number occurrence to the rest of occurrences in a file. Start at the second time the word appears and continue to till the end of the file:
        - `sed 's/pizza/rice/3g' shopping-list.lst`
    - Replacing string on a specific line number:
        - `sed '3 s/pizza/rice/' shopping-list.lst`
    - Replacing string on a range of lines:
        - `sed '1,3 s/pizza/rice/' shopping-list.lst`