# Week Report 7

## cat

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 in the current working directory: `cat games.txt` Display the content of a file with line numbers: `cat -n ~/Documents/games.txt`

## tac

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 current working directory in reverse order: `tac games.txt` Display the content of a file in reverse order with line numbers: `tac -n ~/Documents/games.txt`

## head

The `head` command displays the top n number of lines of a given file. By default, `head` will print the first 10 lines. If more than one file name is given, then data from each file will be preceded by its file name.

**Formula:** `head` + option + file(s)

**Examples**

Display the first 10 lines of a file: `head homework.txt` Display the first 20 lines of a file: `head -20 homework.txt`

## tail

The `tail` command displays the last n number of lines of a given file. By default, `tail` will print the last 10 lines. If more than one file name is given, then data from each file will be preceded by its file name.

**Formula:** `tail` + option + file(s)

**Examples**

Display the last 10 lines of a file: `tail homework.txt` Display the last 20 lines of a file: `tail -20 homework.txt`

## cut

The `cut` command takes a specific section of each line in a file and displays it on screen.

**Formula:** `cut` + option + file(s)

**Examples**

Display a list of all the users in a system: `cut -d ':' -f1 /etc/passwd` Display a list of all the users in a system with their login shell: `cut -d ':' -f1,7 /etc/passwd`

## paste

The `paste` command joins files horizontally in columns

**Formula:** `paste` + option + file(s)

**Examples**

Merge two files: `paste games.txt homework.txt` Merge two files using a different delimiter: `paste -d ':' games.txt homework.txt`

## sort

The `sort` command sorts files. It supports sorting alphabetically, in reverse order, by number, and by month.

**Formula:** `sort` + option + file

**Examples**

Sort a file: `sort groceries.docx` Sort a file in reverse order: `sort -r groceries.docx`

## wc

The `wc` command prints 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 games.txt` Display the number of lines in a file: `wc -l games.txt`

## tr

The `tr` command translates or deletes characters from standard output

**Formula:** standard output | tr + option + set + set

**Examples**

Translate one character to another: `cat program.py | tr '.' ','`

Translate white spaces to tabs: `cat program.py | tr '[:space:]' '/t'`

## diff

The `diff` command compares files and displays the differences between them

**Formula:** `diff` + option + file1 + file2

**Examples**

Display the difference between two files: `diff homework.txt games.txt`

Display the difference between two files in a column format: `diff -y homework.txt games.txt`

## grep

The `grep` command searches text in a given file. It works in a line by line basis.

**Formula:** `grep` + option + search criteria + file(s)

**Examples**

Search any line that contains the word "math" in the given file: `grep 'math' ~/Documents/homework.txt`

Search for all the lines that do not contain the word "English": `grep -v 'English' ~/Documents/homework.txt`

## awk

The `awk` command is a scripting language used for processing and displaying text. `awk` can work with a text file or from standard output.

**Formula:** `awk` + options + {awk command} + file + file to save (optional)

**Examples**

Print the first column of every line of a file: `awk '{print $1}' ~/Documents/homework.txt`

Print the first field of /etc/passwd file: `awk -F: '{print $1}' /etc/passwd`

Print the last field of /etc/passwd file: `awk -F: '{print $NF}' /etc/passwd`

Print the first and last field of /etc/passwd file: `awk -F: '{print $1, '=', $NF}' /etc/passwd`

Print the first and 3 fields with line numbers: `awk -F: '{print NR,$1,$3 }' /etc/passwd`

## sed

The `sed` command is a stream editor that preforms operations on files and standard output. For example, it can search, find and replace, insert, and delete.

**Formula:** `sed` options + sed script + file

**Examples**

Replacing a string in a given file (replacing "pizza" for "rice"): `sed 's/pizza/rice' shopping-list.lst`

Replacing a string on a specific line number: `sed 3 's/pizza/rice' shopping-list.lst`

Replacing a string on a range of lines: `sed '1,3 s/pizza/rice' shopping-list.lst`

Deleting a particular line (line 10): `sed '10d' shopping-list.lst`

Deleting the last line: `sed '$d' shopping-list.lst`