

cat Cat is used for concatenation and displaying the output but is also used simply to display the contents of a file

cat + options + files to display

ex 1: *cat -n ~/Documents/Books/bible/.txt displays the file at that location with numbers indicating the lines of text*

ex 2: *cat -s ~/Documents/Books/bible/.txt displays the supplied file and will suppress blank lines to just one*

tac tac is used to concatenate and display files in reverse order and can be used to simply display a single file line-by-lines in reverse order

tac + option + file to display

ex 1: *tac ~/Documents/Books/bible/.txt displays the lines of the supplied file in reverse order*

head head will display the first 10 lines of a given file. The number of lines shown can be modified using options

head + option + file(s)

ex 1: *head ~/Documents/Books/dracula.txt ~/Documents/Books/bible.txt displays the first 10 lines of the two given files, will print the absolute location at the start of each file's output*

ex 2: *head -5 ~/Documents/Books/dracula.txt displays only the first five lines of the given file*

tail tail will display the last 10 lines of a given file. The number of lines shown can be modified using options

tail + option + file(s)

ex 1: *tail -3 ~/Documents/Books/dracula.txt displays the last 3 lines of a given text*

ex 2: *tail -n5 --lines=10 ~/Documents/Books/dracula.txt starts displaying the file at line 5 and from there it will display 10 lines*

cut cut is used to extract a specific section of each line of a file. It is useful to think of the file as a table and each line of the file represents a row. There is a delimiter used to indicate where the cells of each row exist. *cell 1;cell 2;cell 3* represents three separate cells and the semicolon is the delimiter. Make sure you're specifying the correct delimiter that your files use. You also will want to specify which fields to cut out so this command should always require options.

cut + option + file

ex 1: *cut -d ',' -f3 ~/myfile.csv cuts a file specifying the delimiter as a comma and only displaying field 3 of each line*

ex 2: *cut -sd ';' -f1,3 ~/Documents/Csv/cars.csv cuts fields 1 and 3 out of a file using semicolon as the delimiter and will not display any lines not containing the delimiter. By default cut always displays lines that do not contain a delimiter*

paste paste is used to joining files horizontally in columns, both line ones will be connected to make a new line one and then both line twos will become the new line two etc. You'll likely be using csv files so you'll want to specify the delimiter.

paste + option + files

ex 1: `paste ~/Desktop/animal.csv ~/Desktop/new.txt` *joins the two files together horizontally*

ex 2: `paste -d ',' ~/Desktop/animal.csv ~/Desktop/new.txt` *joins two files together horizontally specifying a delimiter*

sort sort is used to rearrange the lines of a file and can be specified in a number of different ways. By Default lines starting with numbers appear before lines starting with letters and lowercase letter appear before the uppercase version.

sort + option + file

ex 1: `sort -o myfile.txt mySortedFile.txt` *sorts the first file and then saves the output under the second supplied file*

ex 2: `sort -bf myfile.txt` *sorts the given file ignoring leading blanks and treating upper and lowercase with the same priority*

wc wc is used to print certain characteristics of a file such as size, word count, line count

wc + option + file

ex 1: `wc -l myfile.txt` *displays the number of lines in a file*

ex 2: `ec -w myfile.txt` *displays the number of words in a file*

tr tr is used to replace characters from a standard output into new characters. Therefore something must be piped into it from an output

standard output | tr + option + set + set

ex 1: `cat file.txt | tr ',' '.'` *takes the output from the cat command and replaces all commas with periods*

ex 2: `cat file.txt | tr "[:space:]" '\t'` *replaces all the spaces into tabs*

diff displays the differences between two files

diff + option + file1 + file2

ex 1: `diff ~/Desktop/animal.csv ~/Desktop/new.txt` *displays the difference between the given files*

ex 2: `diff -q ~/Desktop/animal.csv ~/Desktop/new.txt` *will only tell you that the files differ without showing the differences*

grep grep will search through a file for a given criteria and then displays the lines with the it occurs. grep is case sensitive unless otherwise specified.

grep + option + search criteria + file(s)

ex 1: `grep -n 'Jerry' file.txt` *displays all the lines containing Jerry with the number line*

ex 2: `grep -v 'Jerry' file.txt` *displays all the line that don't contain 'Jerry'*

ex 3: `grep -w 'erry' file.txt` *specifies that the match must not be part of another word. Normally erry will match all the instances of Jerry, but -w makes it so only erry specifically will come as a match

awk awk is a programming language used for processing and displaying text and can be used to cover much of the functionality of the previous commands

*awk + option + {awk command} + file + file to save (optional)

ex 1: `awk '{print $1}' file.txt` *print the first column of every line of a file*

ex 2: `awk '{FS=";"}{print $1,$4}' ~/Documents/Csv/cereal.csv` *sets the delimiter of the input to a semicolon and prints fields 1 and 4*

ex 3: `awk '{FS=";"}{OFS=","}{print $1,$4}' ~/Documents/Csv/cereal.csv` *sets the input delimiter as a semicolon and then displays fields 1 and 4 with a comma between them as a delimiter

ex 4: `awk 'NR > 20 { print }' ~/Documents/Csv/cars.csv` *display a file starting at line 20*

ex 5: `awk '{print length($0)}' /etc/passwd` *displays the length of each line*

sed sed is stream editor that works on files or standard output. It allows you to edit files without opening them*

sed + options + sed script + file

ex 1: `sed 's/todo/done' checklist.txt` *replaces all instances of todo with done in the given file*

ex 2: `sed '1,5 s/todo/done' checklist.txt` *replaces words within a range of lines*

ex 3: `sed 's/todo/done/3g' checklist.txt` *replace starting after a given number of occurrences of a given string*

ex 4: `sed '5d' checklist.txt` *delete a specific line*

ex 5: `sed 'G' checklist.txt` *insert a blank line after every line*