Filter Text/Processing Commands

- ## cut Command:

Purpose:

Extracts specific columns or fields from a file or input.

Syntax:

cut [OPTIONS] FILE

Example:

cut -f 2,4 filename.txt

Extracts the second and fourth fields from filename.txt.

- ## awk Command:

Purpose:

Powerful text processing tool for pattern scanning and processing.

Syntax:

awk 'pattern { action }' FILE

Example:

awk '{print $1, $NF}' data.txt

Prints the first and last fields of each line in data.txt.

- ## grep/egrep Commands:

Purpose:

Searches for patterns in text.

Syntax:

grep [OPTIONS] PATTERN FILE

Example:

grep "error" log.txt

Displays lines containing the word "error" in log.txt.

- ## sort/uniq Commands:

sort Purpose:

Sorts lines of text files.

sort Syntax:

sort FILE

uniq Purpose:

Filters adjacent identical lines, often used with sort.

uniq Syntax:

uniq [OPTIONS] FILE

Example:

sort data.txt | uniq

Sorts and displays unique lines from data.txt.

- ## wc Command:

Purpose:

Counts lines, words, and characters in a file.

Syntax:

wc [OPTIONS] FILE

Example:

wc -l filename.txt

Counts the number of lines in filename.txt.

- ## diff Command:

Purpose:

Compares and shows the differences between two files.

Syntax:

diff FILE1 FILE2

Example:

diff file1.txt file2.txt

Displays differences between file1.txt and file2.txt.

- ## cmp Command:

Purpose:

Compares two files byte by byte.

Syntax:

cmp FILE1 FILE2

Example:

cmp file1.txt file2.txt

Compares file1.txt and file2.txt and reports if they differ.

These commands are fundamental for text processing, searching, and comparing files in a Unix/Linux environment, providing versatile tools for data manipulation and analysis.

References:

https://www.geeksforgeeks.org/cut-command-linux-examples/

https://www.geeksforgeeks.org/awk-command-unixlinux-examples/

https://www.geeksforgeeks.org/diff-command-linux-examples/

https://www.geeksforgeeks.org/cmp-command-in-linux-with-examples/