Notes 8

awk

- text processing and data extraction. It allows users to manipulate structured data and perform operations like filtering, formatting, and calculatins on text files or output from other commands.

Usage/Formula

- awk 'pattern { action }' filename
  - pattern: specifies a condition to match lines in a file
  - { action }: defines what to do with matching lines
  - filename: is the input file to process

Examples:

- Printing specific Columns
  - awk '{print $1}' data.txt
- Filtering Lines Based on aa Condition
  - awk '$2 > 25 {print $1, $2}' data.txt
- Performing Calculations
  - awk '{sum += $2} END {print "Total Age:", sum}' data.txt

sed

- performs basic text transformations and filtering on an input strwam or a file. It operates line by line and is often used for automated editing of files or text manipulation

Usage/Formula

- sed 'command' filename
  - 'command': specifies the operation to perform
  - filename: file to be processed

Examples

- Find and replace text
  - sed 's/Hello/Hi/g' file.txt
- Delete specific lines
  - sed '/error/d' file.txt
- Print only specific lines
  - sed -n '1,3p' file.txt

less

- used for viewing text files without opening an editor. it provides smooth navigation through long files with features like scrolling, searching, and jumping to specific sections.

Usage/Formula

- less filename
    - filename: the file you want to view

Examples

- Viewing a file
    - less file.txt
- Viewing command output
    - ls -l | less

>

- operator redirects standard output (stdout) to a file. If the file already exists, its contents are overwritten

Usage/Formula

- command > filename
    - command: generates output
    - filename: the destination file

Examples

- Redirect output to a file
    - ls > output.txt
- Overwritting a file
    - echo "Hello, World!" > myfile.txt
- Redirecting error messages
    - command 2> errors.txt

>>

- redirects standard output (stdout) and appends it to an existing file. IF the file does not exist, it is created.

Usage/Formula

- command >> filename
    - command:generates output
    - filename: the destination file, which will be appended

Example

- Appending output to a file
    - ls >> output.txt
- Appending text to a file
    - echo "New entry added!" >> notes.txt

- Logging command output
    - command 2>> error_log.txt

|

- enables command output redirection by sending the output form one command into another command. This eliminates the need for temporary files and streamlines workflows.

Usage/Formula

- command1 | command2
    - command1: produces output
    - command2: recieves and processes that output

Examples

- Viewing long outputs
    - ls -l | less
- Filtering Output
    - cat logfile.txt | grep "error"
- counting lines in a file
    - cat file.txt | wc -l