

DAY 6

The **awk** command is a powerful tool for text processing in Unix-like operating systems. It's a scripting language that's designed for processing structured text like CSV, TSV, or log files.

Text Processing: awk is often used to process text files. For example, `awk '{print $1}' filename` will print the first field (column) of each line in the file.

Pattern Scanning: awk can scan a file line by line for a pattern. When a line matches one of the patterns, awk performs specified actions on that line. For example, `awk '/pattern/ {print $0}' filename` will print the lines that match the pattern.

Text Transformation: awk can transform the data in a text file. For example, `awk '{$1=""; print $0}' filename` will remove the first field from each line in the file.

Calculations: awk can perform mathematical calculations. For example, `awk '{sum+=$1} END {print sum}' filename` will sum up the values of the first field in the file.

Generating Reports: awk can generate reports based on the data in a text file. For example, `awk '{count[$1]++} END {for (word in count) print word, count[word]}' filename` will print each unique word in the first field and its count.

The **sed** command, short for Stream Editor, is a powerful utility in Unix-like operating systems. It's primarily used for text manipulation and transformation.

Find and Replace Text: The most common use of sed is to find and replace text in a file. For example, `sed 's/foo/bar/g' filename` will replace all occurrences of 'foo' with 'bar' in the specified file.

Delete Lines: sed can be used to delete lines in a file that match a specific pattern. For example, `sed '/pattern_to_match/d' filename` will delete all lines containing 'pattern_to_match' from the file.

Insert and Append Text: sed can insert text before a line or append text after a line in a file. For example, `sed '3i\This is the inserted text.' filename` will insert 'This is the inserted text.' before the third line of the file.

Modify Lines: sed can change any text in a file based on a regular expression. For example, `sed 's/^/prefix_/' filename` will add 'prefix_' at the beginning of every line in the file.

Print Lines: sed can print only those lines that match a specific pattern. For example, `sed -n '/pattern_to_match/p' filename` will print only the lines that contain 'pattern_to_match'.

`#!/bin/bash` is known as a shebang or hashbang. It is used in scripts to indicate an interpreter for execution under UNIX and UNIX-like operating systems.

`#!` is a two-byte magic number, a special marker that designates a file type, or in this case an executable shell script (starting with `#!` is actually interpreted as a directive to the program loader to run the rest of the text file as a program).

`/bin/bash` is the path to the interpreter that should be used to execute the script. In this case, it's the Bash shell.

So, `#!/bin/bash` at the start of a script tells the system that this script should be executed with the Bash shell, even if the shell is not the current shell for the user.

For CPU utilization, it uses the `top` command with `-bn1` option to run `top` once in batch mode and then `grep` to find the line containing "Cpu(s)". The `sed` command is used to extract the idle CPU percentage and `awk` is used to subtract it from 100 to get the CPU utilization.

For memory utilization, it uses the `free` command with `-m` option to display the amount of free and used memory in the system in MB. `awk` is used to calculate the percentage of used memory.

For disk utilization, it uses the `df` command with `-h` option to report file system disk space usage in human-readable format. `awk` is used to print the percentage of disk usage for the root directory ("/").

DAY 6 - END