Working With Linux Files

A Hands-On Guide for Beginners

A practical guide to essential Linux commands for manipulating text files from the terminal

Introduction

In Linux, many configuration and log files are stored as plain text. Mastering powerful tools for editing and analyzing text files is essential. This document introduces various methods and commands for text manipulation.

1. Displaying Files

1.1 cat – Display the Entire Content of a File

The cat command prints the contents of a file to the terminal:

```
→ cat file.txt
```

Limitations:

Not ideal for large files, as the entire content is displayed at once.

1.2 head – Show the First Few Lines

By default, head displays the first 10 lines of a file:

```
→ head file.txt
```

To specify the number of lines:

```
→ head -20 file.txt                    # Shows the first 20 lines
```

1.3 tail – Show the Last Few Lines

Similar to head, but displays the last lines of a file. Useful for monitoring logs:

```
→ tail -f /var/log/syslog              # Live updates of new entries
```

2. Line Numbering

Use nl to display a file with line numbers:

```
→ nl file.txt
```

3. Filtering Text with grep

3.1 Search for Specific Words

grep is used to find lines containing a specific word:

```
→ grep "error" log.txt                 # Displays all lines containing "error"
```

To ignore case sensitivity:

→ grep -i "error" log.txt # -i, --ignore-case

3.2 Show Lines That DO NOT Contain a Specific Word

→ grep -v "WARNING" log.txt # Displays all lines except those containing "WARNING"

3.3 Recursive Search in Multiple Files

→ grep -r "root" /etc/

4. Find and Replace with sed

sed is used to find and replace text in files.

4.1 Basic Word Replacement

→ sed 's/old/new/g' file.txt

g means **global**, replacing all occurrences in the file.

4.2 Modify a File Directly

→ sed -i 's/old/new/g' file.txt

4.3 Replace Text in a Specific Line

Replace "test" only in line 10:

→ sed '10s/test/replaced/g' file.txt

5. Viewing Files Page by Page

5.1 more – Simple Pagination

→ more large_file.txt

Navigation:

SPACE – Next page **q** – Quit

5.2 less – More Control

→ less large_file.txt

Navigation:

/search_term – Search for a word

n – Jump to the next match

q – Quit

6. Advanced Text Processing with awk

6.1 Extract Specific Columns

If a file contains space- or tab-separated values:

→ awk '{print $1, $3}' file.txt # Displays the first and third column

6.2 Display Only Specific Rows

→ awk '$3 > 50 {print $0}' data.txt

This shows only rows where the value in column 3 is greater than 50.

7. Editing Multiple Files

7.1 Merging Files

→ cat file1.txt file2.txt > merged.txt

7.2 Find and Replace in Multiple Files

Replace a word in all `.txt` files in a directory:

→ sed -i 's/oldword/newword/g' *.txt

8. Sorting and Removing Duplicates

8.1 Sorting

→ sort file.txt

8.2 Remove Duplicate Lines

→ sort file.txt | uniq

8.3 Count Frequency of Words

→ sort file.txt | uniq -c | sort -nr

This shows which words appear most frequently.

9. Efficient Log File Monitoring

9.1 Live Monitoring of Logs

→ tail -f /var/log/syslog

With grep, you can filter for errors:

→ tail -f /var/log/syslog | grep "error"

Conclusion

Working with text files in Linux is an essential skill for system administrators, developers, and security professionals. Using `grep`, `sed`, `awk`, and `sort`, you can efficiently process and analyze data. Regular practice will help you master these techniques and use them effectively.