

DAY 4x/14 LEARNING ABOUT LINUX

Exploring SED - The Stream Editor

In the realm of Unix and Linux, text processing stands as a fundamental task, essential for both system administration and programming. Among the suite of powerful utilities available, **sed**, the stream editor, occupies a pivotal role. This utility, designed for filtering and transforming text in a stream, is an indispensable tool for performing simple to moderately complex editing operations directly from the command line or within shell scripts.

UNDERSTANDING sed

sed operates by processing input text line by line, applying a specified script of commands to each line. These commands can insert, delete, or substitute text according to patterns or line numbers. Unlike text editors where modifications are made directly within files, sed applies changes to the input stream and outputs the result, leaving the original file unchanged unless redirection is used.

I. SYNTAX AND COMMAND STRUCTURE

The basic syntax of sed is:

```
sed [options] 'command' file
```

Where the command includes an action to be performed and optionally addresses indicating which lines the action should apply to.

II. COMMON USE CASES

- **Substitution:** Perhaps the most frequently utilized operation in sed scripts, substitution allows replacing text that matches a regular expression.

Example: To replace "cat" with "dog" in file.txt:

```
sed 's/cat/dog/' file.txt
```

- **Deletion:** sed can remove lines from the input based on a match.

Example: To delete lines containing "delete me" in file.txt:

```
sed '/delete me/d' file.txt
```

- **Insertion and Appending:** Inserting or appending text before or after a matched line or pattern.

Example: To insert a header at the beginning of a file:

```
sed '1i\This is the header' file.txt
```

III. ADVANCED MANIPULATIONS

- **In-place Editing:** Using the **-i** option, sed can modify files in place, though it's crucial to use this with caution to avoid accidental data loss.

Example: To replace "old" with "new" in file.txt, modifying the file directly:

```
sed -i 's/old/new/' file.txt
```

- **Multi-command Scripts:** sed supports executing multiple editing commands in sequence, separated by semicolons or placed in a script file.

Example: To add line numbers to a file:

```
sed '=' file.txt | sed 'N; s/\n/ /'
```

IV. PRACTICAL EXAMPLES

1. **Trimming Whitespace:** Removing leading and trailing whitespace from each line in a file.

```
sed 's/^[ \t]*//;s/[ \t]*$//' file.txt
```

2. **Selective Line Editing:** Editing lines that match a specific pattern or fall within a line range.

```
sed '/pattern/s/find/replace/' file.txt
```

3. **Commenting Out Lines Matching a Pattern:** Adding a "#" at the beginning of lines containing "DEBUG".

```
sed '/DEBUG/s/^/#/' file.txt
```

4. **Extracting Specific Lines:** Displaying only lines 10 through 20 of a file.

```
sed -n '10,20p' file.txt
```

Concluding Thoughts

sed exemplifies the power and flexibility of Unix text processing tools, providing a robust mechanism for automating editing tasks across diverse text formats and environments.

Homework TC 4x

I. Homework 1: Getting Started with sed

Objective: Familiarize yourself with basic sed operations.

1. **Simple Substitution:** Replace all occurrences of "apple" with "orange" in the file fruits.txt.

2. **Line Deletion:** Remove all blank lines from the file spaces.txt.

```
# Simple Substitution
sed 's/apple/orange/g' fruits.txt

# Line Deletion
sed '/^$/d' spaces.txt
```

II. Homework 2: Pattern Matching and Line Operations

Objective: Practice using sed for selective editing based on patterns.

1. **Selective Substitution:** In report.txt, replace "failed" with "passed" only in lines containing "Test Case".
2. **Deleting Lines Containing a Pattern:** Remove all lines that contain the word "obsolete" from products.txt.

```
#Selective Substitution
sed '/Test Case/s/failed/passed/g' report.txt

#Deleting Lines Containing a Pattern
sed '/obsolete/d' products.txt
```

III. Homework 3: In-Place Editing and Multiline Processing

Objective: Learn to modify files in place and perform operations affecting multiple lines.

1. **In-Place File Editing:** Without creating a backup, change all instances of "http:" to "https:" in urls.txt.
2. **Extracting a Range of Lines:** Using sed, extract lines 10 through 20 from log.txt and write them to extracted_log.txt.

```
#In-Place File Editing
sed -i 's/http:/https:/g' urls.txt

#Extracting a Range of Lines
sed -n '10,20p' log.txt > extracted_log.txt
```

IV. Homework 4: Advanced Text Manipulation

Objective: Apply sed to perform complex text transformations and formatting.

1. **Appending and Inserting Text:** For each line in chapters.txt starting with "Chapter", insert a new line with "---" above it and append "-----" below it.
2. **Complex Pattern Replacement:** In codes.txt, replace all occurrences of digits (0-9) with their corresponding word (e.g., "1" to "one"). This task tests your ability to use back-references and grouping in sed.

```
#Appending and Inserting Text
sed '/^Chapter/{i\
-----
a\
```

```
-----  
}' chapters.txt  
  
#Complex Pattern Replacement (simplified example for replacing "1" with "one")  
sed 's/1/one/g' codes.txt
```

V. Homework 5: Scripting with sed

Objective: Create sed scripts to handle multi-command editing tasks.

1. **Multi-Command Scripting:** Write a sed script to perform the following operations on article.txt: First, capitalize all occurrences of the word "linux"; then, delete all lines containing "draft".
2. **Selective Editing Script:** Create a sed script to comment out lines starting with "DEBUG:" and, in the same pass, replace "TODO" with "DONE" in code.txt.

```
#Multi-Command Scripting  
  
#Execute with: sed -f sed_script.sed article.txt  
  
# sed_script.sed  
  
/linux/s//Linux/g  
  
/draft/d  
  
#Selective Editing Script  
  
#Execute with: sed -f script.sed code.txt  
  
# script.sed  
  
/^DEBUG:/s/^/#/  
  
s/TODO/DONE/g
```

Submission Guidelines

- Ensure your scripts are well-commented to explain your logic and approach.
- Test your scripts with sample data to verify correctness.
- Submit (in the forum) your SED scripts along with a brief report describing your solution strategy for each task. Also export history and upload it as usual.