Assignment :2

Task : 3

3. Text Processing Commands:

a. Task: Search and replace text in a file - Use the sed command to search for and replace specific text in a text file. Demonstrate the use of regular expressions.

Certainly! The sed command is a powerful tool for text processing and can be used to search and replace text in a file. Here's an example of how to use sed to perform a search and replace operation with regular expressions:

```
sed 's/search_pattern/replacement_text/g' input_file > output_file
```

Here's a breakdown of the command:

sed: The command for stream editor.

s: Stands for substitute.

search_pattern: Replace this with the pattern you want to search for. You can use regular expressions here.

replacement_text: Replace this with the text you want to replace the matched pattern with.

g: Stands for global, which means replace all occurrences on each line. If you omit this, only the first occurrence on each line will be replaced.

input_file: Replace this with the path to the input file.

output_file: Replace this with the path to the output file.

Example :

For example, let's say you have a file named "example.txt" with the following content:

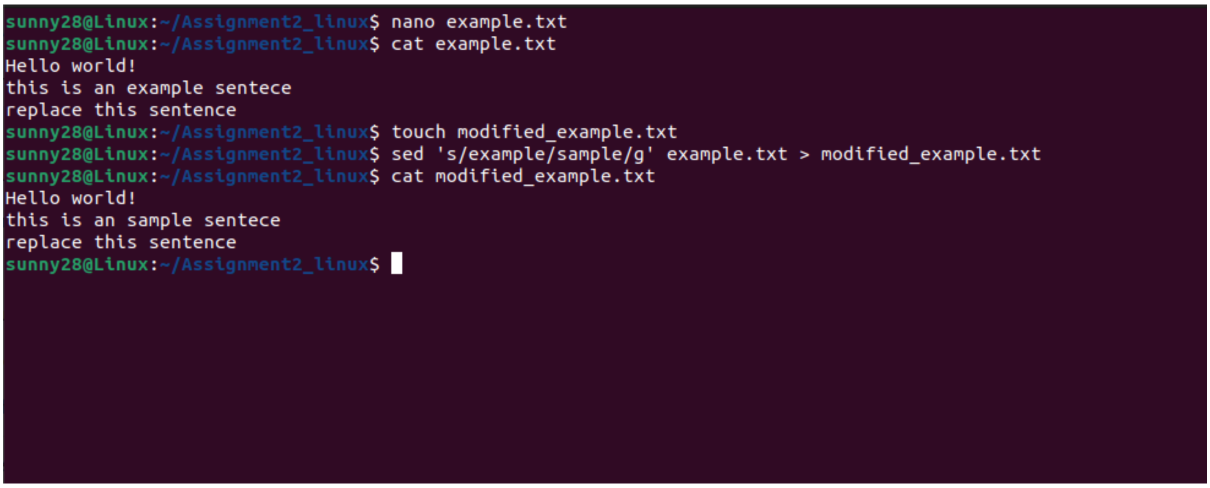
Hello world!

this is an example sentence.

replace this text.

And you want to replace "example" with "sample", you would run:

```
sed 's/example/sample/g' example.txt > modified_example.txt
```



```
sunny28@Linux:~/Assignment2_linux$ nano example.txt
sunny28@Linux:~/Assignment2_linux$ cat example.txt
Hello world!
this is an example sentece
replace this sentence
sunny28@Linux:~/Assignment2_linux$ touch modified_example.txt
sunny28@Linux:~/Assignment2_linux$ sed 's/example/sample/g' example.txt > modified_example.txt
sunny28@Linux:~/Assignment2_linux$ cat modified_example.txt
Hello world!
this is an sample sentece
replace this sentence
sunny28@Linux:~/Assignment2_linux$ 
```

After running this command, the content of "modified_example.txt" would be:

Hello world!

this is an sample sentence.

replace this text.

b. Task: Extract and manipulate data from a CSV file - Use awk to extract and manipulate data from a CSV file. Show how to perform operations like summing columns or filtering data based on criteria.

Certainly! awk is a powerful tool for text processing and is particularly useful for working with structured data like CSV files. Here's an example of how you can use awk to extract and manipulate data from a CSV file:

Let's assume you have a CSV file named "data.csv" with the following content:

Name, Age, Salary

sunny, 25, 50000

Bob, 30, 60000

Alice, 35, 70000

Example 1: Summing Columns

To calculate the sum of a specific column (e.g., the Salary column), you can use awk as follows:

awk -F',' '{ sum += $3 } END { print "Total Salary: " sum }' data.csv

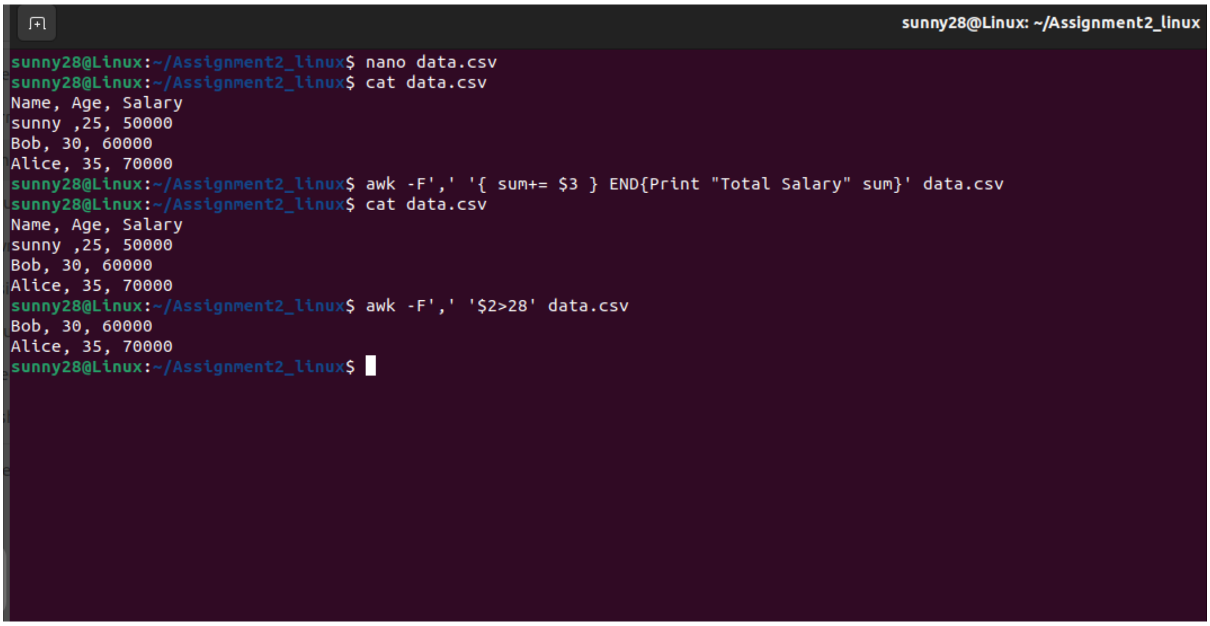
This command uses -F',' to specify that the field separator is a comma. It then adds up the values in the third column (Salary) and prints the total at the end.

Example 2: Filtering Data Based on Criteria

To filter data based on a specific criterion (e.g., only display rows where Age is greater than 28), you can use:

awk -F',' '$2 > 28' data.csv

This command checks if the value in the second column (Age) is greater than 28 and prints the entire row if the condition is true.



```
sunny28@Linux:~/Assignment2_linux$ nano data.csv
sunny28@Linux:~/Assignment2_linux$ cat data.csv
Name, Age, Salary
sunny ,25, 50000
Bob, 30, 60000
Alice, 35, 70000
sunny28@Linux:~/Assignment2_linux$ awk -F',' '{ sum+= $3 } END{Print "Total Salary" sum}' data.csv
sunny28@Linux:~/Assignment2_linux$ cat data.csv
Name, Age, Salary
sunny ,25, 50000
Bob, 30, 60000
Alice, 35, 70000
sunny28@Linux:~/Assignment2_linux$ awk -F',' '$2>28' data.csv
Bob, 30, 60000
Alice, 35, 70000
sunny28@Linux:~/Assignment2_linux$
```