



Awk Script: Counting Characters, Words, and Lines in a File

Learn how to use Awk, a powerful text processing tool, to analyze file data. Discover the importance of character, word, and line counts in file analysis.

R by Ravilla Sravani

Introduction

- Why counting characters, words, and lines is crucial in file analysis
- The purpose of the Awk script
- What is Awk and how it compares to sed

Awk: A versatile tool for manipulating file data, similar to sed.

NF: Command to count the number of fields in a record.

NR: Command to count the number of records in a file.

Mv: Command to move data between files.

Commands and Syntax

- Explore the commands with syntax and examples
- Write the code snippet below
- BEGIN{print "record.\t characters \t words"}

#BODY section

```
{  
len=length($0)  
total_len =len  
print(NR":\t",len,":\t",NF,$0)  
words=NF  
}  
END{  
print("\n total")  
print("characters :\t" total len)  
print("lines :\t" NR)  
}
```

Sample output: Student@ubuntu:~\$ awk -f cnt.awk ff1 Record words 1: 5: 1hello Total Characters:5 Lines:1

Counting Characters

- Implementing the Awk script to count characters in a file
- Formatting and display preferences for character count output

Counting Words

- Implementing the Awk script to count words in a file
- Handling different word delimiters and special cases

Counting Lines

- Implementing the Awk script to count lines in a file
- Managing empty lines and end-of-file conditions

Conclusion

- Summary of Awk script functionality for character, word, and line counts
- Importance of efficient file analysis using Awk