SED: A Stream Editor

Prashant K. Sharma
Diptesh Kanojia

e-Yantra Team
ERTS Lab, IIT Bombay

IIT Bombay
April 12, 2021

Agenda for Discussion

1. Sed: What and Why ?
   - A stream Editor

2. Sed: How?
   - Basic Usage
   - Printing lines
   - Address Ranges
   - Deleting Text
   - Substituting Text

Introduction

SED is used to manipulate the text in Linux. It's called a stream editor as it performs the editing operations line-by-line on the information coming from standard input or a file.

Introduction

SED is used to manipulate the text in Linux. It's called a stream editor as it performs the editing operations line-by-line on the information coming from standard input or a file.

Command Syntax

```
sed [OPTIIONS] commands [INPUTFILE]
```

Introduction

SED is used to manipulate the text in Linux. It's called a stream editor as it performs the editing operations line-by-line on the information coming from standard input or a file.

Command Syntax

```
sed [OPTIIONS] commands [INPUTFILE]
```

Let's look at basic usage by performing specific tasks using sed.
NOTE: All the task consider the file `players.csv`, which will be shared with you.

Basic Usage

sed operates on a steam of text that it reads from an input file or standard input (STDIN). Thus, sed can be used to edit a file that you have created or output of another command.

Basic Usage

sed operates on a steam of text that it reads from an input file or standard input (STDIN). Thus, sed can be used to edit a file that you have created or output of another command.

How to view the contents of the input file ?

Basic Usage

sed operates on a steam of text that it reads from an input file or standard input (STDIN). Thus, sed can be used to edit a file that you have created or output of another command.

How to view the contents of the input file ?

Command Syntax

```
sed '' players.csv
```

Basic Usage

sed operates on a steam of text that it reads from an input file or standard input (STDIN). Thus, sed can be used to edit a file that you have created or output of another command.

How to view the contents of the input file ?

Command Syntax

```
sed '' players.csv
```

Output

```
You should see all the content of players.csv displayed on
the screen.
```

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to print every line of input to `sed` ?

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to print every line of input to `sed` ?

Use of `print` i.e `p` command

```
sed ’p’ players.csv
```

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to print every line of input to `sed` ?

Use of `print` i.e `p` command

```
sed ’p’ players.csv
```

Output

You should see every line of players.csv printed twice. Why? sed prints each line by default and you are explicitly telling it to print lines with p command.

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to suppress the automatic printing of every line of input?

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to suppress the automatic printing of every line of input?

Use of -n option

```
sed -n 'p' players.csv
```

Printing Lines

sed when passed with an input without any specific command directly prints the results onto the standard output.

How to suppress the automatic printing of every line of input?

Use of -n option

```
sed -n 'p' players.csv
```

Output

```
You should see every line of players.csv printed once.
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print the 2nd line of the input csv file ?

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print the 2nd line of the input csv file ?

Use of p command

```
sed -n '2p' players.csv
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print the 2nd line of the input csv file ?

Use of p command

```
sed -n '2p' players.csv
```

Output

```
"Adam Donachie", "BAL", "Catcher", 74, 180, 22.99
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print first 5 lines of the input csv file ?

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print first 5 lines of the input csv file ?

Use of p command

```
sed -n '1,5p' players.csv
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print first 5 lines of the input csv file ?

Use of p command

```
sed -n '1,5p' players.csv
```

Output

```
You should see first first 5 lines, including the header
of players.csv file.
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print every other line of the input csv file ?

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print every other line of the input csv file ?

Use of p command

```
sed -n '1~2p' players.csv
```

Using Address Ranges

To target specific parts of a text stream, you can make use of addresses. Either a specific line or range of lines can be targeted.

How to print every other line of the input csv file ?

Use of p command

```
sed -n '1~2p' players.csv
```

Output

```
You should see alternate lines of players.csv file printed
on your screen.
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file ?

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file ?

Use of d command

```
sed '1~2d' players.csv
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file ?

Use of d command

```
sed ’1~2d’ players.csv
```

Output

```
You should see every line except the alternate lines of
players.csv file printed on your screen.
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** ?

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** ?

Use of d command

```
sed -i '1~2d' players.csv
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** ?

Use of d command

```
sed -i '1~2d' players.csv
```

Output

```
You won't see any output on the terminal, as inplace
deletion is done.
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** with backup ?

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** with backup ?

Use of d command

```
sed -i.bak '1~2d' players.csv
```

Deleting Text

To do deletion from the text, change the previous used **p** command (for printing lines) with **d** command (for deleting lines)

How to delete every other line of the input csv file **(in-place)** with backup ?

Use of d command

```
sed -i.bak '1~2d' players.csv
```

Output

```
You won't see any output on the terminal, as inplace
deletion is done. This creates a backup file with .bak
extension
```

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace comma(,) with semicolon(;) of the input csv file ?

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace comma(,) with semicolon(;) of the input csv file ?

Use of s command

```
sed 's/,/;/' players.csv
```

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace comma(,) with semicolon(;) of the input csv file ?

Use of s command

```
sed ’s/,/;/’ players.csv
```

Output

You should see the first comma(,) of every line is replaced with semicolon(;). But, not all commas are replaced!

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace all commas(,) with semicolon(;) of the input csv file ?

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace all commas(,) with semicolon(;) of the input csv file ?

Use of s command with g flag

```
sed ’s/,/;/g’ players.csv
```

Substituting Text

sed can be utilised to search for patterns via regular expressions and then replace the found text with desired text.

How to replace all commas(,) with semicolon(;) of the input csv file ?

Use of s command with g flag

```
sed 's/,/;/g' players.csv
```

Output

```
You should see all commas(,) of every line is replaced
with semicolon(;).
```

References

- sed, a stream editor

Thank You!

Author: Diptesh Kanojia
Contributor: Prashant Sharma

Post your queries at: resources@e-yantra.org