sed: THE STREAM EDITOR

CONTENT

- grep filter
- Different options
- Regular expressions
  - Basic regular expressions
  - Extended regular expressions

CONTENT

- sed to edit an input stream and understand its addressing mechanism
- Line addressing
- Using multiple instructions
- Context addressing
- Writing selected lines to a file
- Text editing
  - Inserting, changing and deleting lines

INTRODUCTION

- sed is a multipurpose tool which combines the work of several filters
- sed uses **instructions** to act on text. An instruction combines an **address** for selecting lines, with an **action** to be taken on them.

  sed options ‘address action’ file(s)

- sed supports only the BRE set

Address specifies either one line number to select a single line or a set of two lines, to select a group of contiguous lines

action specifies print, insert, delete, substitute the text

LINE ADDRESSING

sed ‘3q’ emp.lst

Just similar to *head –n 3 emp.lst*. Selects first three lines and quits

sed –n ‘1,2p’ emp.lst

p prints selected lines as well as all lines. To suppress this behavior, we use –n whenever we use p command

sed –n ‘$p’ emp.lst

Selects last line of the file

sed –n ‘9,11p’ emp.lst

Selecting lines from anywhere of the file, between lines from 9 to 11

sed –n ‘1,2p

7,9p

$p’ emp.lst

Selecting multiple groups of lines

sed –n ‘3,$!p’ emp.lst

Negating the action, just same as 1,2p

USING MULTIPLE INSTRCUTIONS

- There is adequate scope of using the –e and –f options whenever sed is used with multiple instructions

  sed –n –e ‘1,2p’ –e ‘7,9p’ –e ‘$p’ emp.lst

Let us consider,

cat instr.fil

1,2p

7,9p

$p

- -f option to direct the sed to take its instructions from the file

  sed –n –f instr.fil emp.lst
- We can combine and use –e and –f options as many times as we want

  sed –n –f instr.fil1 –f instr.fil2 emp.lst

sed –n –e ‘/saxena/p’ –f instr.fil1 –f instr.fil2 emp.lst

CONTEXT ADDRESSING

- We can specify one or more patterns to locate lines

  sed –n ‘/director/p’ emp.lst

- We can also specify a comma-separated pair of context addresses to select a group of lines

  sed –n ‘/dasgupta/,/saxena/p’ emp.lst

- Line and context addresses can also be mixed

  sed –n ‘1,/dasgupta/p’ emp.lst

Using regular expressions

- Context addresses also uses regular expressions

Sed –n ‘/[aA]gg*[ar][ar]wal/p’ emp.lst

Selects all agarwals

Sed –n ‘/sa[kx]s*ena/p

/gupta/p’ emp.lst

Selects saxenas and gupta

- We can also use ^ and $, as part of the regular expression syntax

  sed –n ‘/50…..$/p’ emp.lst

Selects all people born in the year 1950

WRITING SELECTED LINES TO A FILE

- We can use w command to write the selected lines to a separate file

  sed –n ‘/director/w dlist’ emp.lst

Saves the lines of directors in *dlist* file

- sed –n ‘/director/w dlist
  /manager/w mlist
  /executive/w elist’ emp.lst

Splits the file among three files

sed –n ‘1,500w foo1

501,$w foo2’ foo.main

Line addressing also. Saves first 500 lines in foo1 and the rest in foo2

TEXT EDITING

- Sed supports inserting (i), appending (a), changing (c) and deleting (d) commands for the text

```
$ sed ‘1i\
> #include <stdio.h>\
> #include <unistd.h>
> ’foo.c > $$
```

Will add two include lines in the beginning of foo.c file. Sed identifies the line without the \ as the last line of input. Redirected to $$ temporary file

- This technique has to be followed when using the a and c commands also
- To insert a blank line *after* each line of the file is printed (double spacing text), we have

```
sed ‘a\

’ emp.lst
```

Deleting lines (d)

sed ‘/director/d’ emp.lst > olist or

sed –n ‘/director/!p’ emp.lst > olist

Selects all lines except those containing *director*, and saves them in *olist*

Note that –n option not to be used with d

SUMMARY

- sed – A multipurpose toolkit and can combine several filters
- Line addressing
- Using multiple instructions
- Context addressing
- Writing selected lines to a file
- Text editing
  - Inserting, appending, changing and deleting lines

- THANK YOU