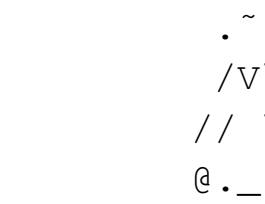
List of Slides

– General Linux 1 –
Search text files using regular expressions []
(Linux Professional Institute Certification)

a



by: geoffrey robertson
geoffrey@zip.com.au

$Id: gl1.103.7.slides.tex,v 1.2 2003/05/30 05:11:28 waratah Exp $

GNU & Unix Commands

`sed`—stream editor

1.103.1 Work on the command line

1.103.2 Process text streams using filters

1.103.3 Perform basic file management

1.103.4 Use streams, pipes, and redirects

1.103.5 Create, monitor, and kill processes

1.103.6 Modify process execution priorities

1.103.7 **Search text files using regular expressions**

Search text files using regular expressions

Objective

The candidate should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.

Search text files using regular expressions

Key files, terms, and utilities

`grep`

`regexp`

`sed`

Search text files using regular expressions

Resources of interest

Fun with Regular Expressions by Adrian J. Chung

`http://thelinuxgurus.org/regexp.html`


grep---

regexes---

sed—stream editor

- `sed` is a non-interactive, or stream oriented editor

sed—stream editor

- `sed` is a non-interactive, or stream oriented editor
- `sed` works in a similar fashion to any text filter, typically taking the lines from a file, filtering them and sending them to STDIO

sed—stream editor

- `sed` is a non-interactive, or stream oriented editor
- `sed` works in a similar fashion to any text filter, typically taking the lines from a file, filtering them and sending them to STDIO
- Example: replace “teh” with “the”

```
$ sed s/teh/the/g my_file.txt ↩
```

sed—stream editor

- `sed` is a non-interactive, or stream oriented editor
- `sed` works in a similar fashion to any text filter, typically taking the lines from a file, filtering them and sending them to STDIO
- Example: replace “teh” with “the”

  ```
  $ sed s/teh/the/g my_file.txt ↩
  ```

- The original file is not touched by sed

sed—stream editor

- `sed` is a non-interactive, or stream oriented editor
- `sed` works in a similar fashion to any text filter, typically taking the lines from a file, filtering them and sending them to STDIO
- Example: replace “teh” with “the”

```
$ sed s/teh/the/g my_file.txt ↩
```

- The original file is not touched by sed
- Save the results:

```
$ sed s/teh/the/g old.txt > new.txt↩
```

sed—stream editor

Calling sed

- sed one liners:

```
$ sed [opts] 'sed-cmds' input-file(s) ↩
```

sed—stream editor

Calling sed

- sed one liners:

```
$ sed [opts] 'sed-cmds' input-file(s) ↩
```

- sed using a script file:

```
$ sed [opts] -f script-file input-file(s) ↩
```

sed—stream editor

Calling sed

- sed one liners:

```
$ sed [opts] 'sed-cmds' input-file(s) ↩
```

- sed using a script file:

```
$ sed [opts] -f script-file input-file(s) ↩
```

- Script with a sed shebang:

```
$ cat script.sed  ↩
#!/bin/sed -f
...
```

sed—stream editor

The sed options

-n No print. The default is to print all lines plus lines selected with the p command.

sed—stream editor

The sed options

-n No print. The default is to print all lines plus lines selected with the p command.

-e The next command is an edit command; used for multiple edits.

sed—stream editor

The sed options

-n No print. The default is to print all lines plus lines selected with the p command.

-e The next command is an edit command; used for multiple edits.

-f sed commands are in a file.

sed—stream editor

Finding text using sed

sed—stream editor

Finding text using sed

- Using line numbers: singly or in a range.

sed—stream editor

Finding text using sed

- Using line numbers: singly or in a range.
- Using Regular Expressions.

sed—stream editor

Finding text using sed

- Using line numbers: singly or in a range.
- Using Regular Expressions.
- Examples:

  x Where x is a line number

  x,y In a range of lines, from x to y

  /pattern/ Where pattern is a regex

  /pattern/pattern/ Choice of patterns

  /pattern/,x Look for the pattern on this line

  x,/pattern/ Look only at line x for the pattern

  x,y! Not lines x to y

sed—stream editor

Basic sed editing commands

p Print the matched lines

= Display the line number of the file

a \ Append the text after the addressed line

i \ Insert new text after the addressed line

d Delete addressed lines

c Replace addressed text with new text

s Substitute pattern with replacement pattern

r Read text from another file

sed—stream editor

Basic sed editing commands

w Write text to file

q Quit after first pattern has been matched, or just quit

l Show control characters in their octal ASCII equivalent

() Group a series of commands to be performed only on addressed lines

n Read the next line of text from another file and append it

g Paste the contents of pattern2 into pattern1

y Translate characters

n Append next input line; this allows pattern matching across two lines

sed—stream editor

sed examples

- Print line 3 only:

```
$ sed -n '3p' foo.txt ↩
```

sed—stream editor

sed examples

- Print line 3 only:

```
$ sed -n '3p' foo.txt ↩
```

- Print lines 5 through 8:

```
$ sed -n '5,8p' foo.txt ↩
```

sed—stream editor

sed examples

- Print line 3 only:

```
$ sed -n '3p' foo.txt ↩
```

- Print lines 5 through 8:

```
$ sed -n '5,8p' foo.txt ↩
```

- Print lines with *Fred* in them:

```
$ sed -n '/fred/p' foo.txt  ↩
```

sed—stream editor

sed examples

- Print line 3 only:

```
$ sed -n '3p' foo.txt ↩
```

- Print lines 5 through 8:

```
$ sed -n '5,8p' foo.txt ↩
```

- Print lines with *Fred* in them:

```
$ sed -n '/fred/p' foo.txt  ↩
```

- Search for *Fred* only on line 4:

```
$ sed -n '4,/Fred/p' foo.txt ↩
```

sed—stream editor

sed examples

- Print lines containing $100:

```
$ sed -n '/\$100/p' foo.txt
```

sed—stream editor

sed examples

- Print lines containing $100:

```
$ sed -n '/\$100/p' foo.txt
```

- Print file 10th line to the last:

```
$ sed -n '10,$p' ↩
```

sed—stream editor

sed examples

- Print lines containing $100:

```
$ sed -n '/\$100/p' foo.txt
```

- Print file 10th line to the last:

```
$ sed -n '10,$p' ↩
```

- Print lines with ing's in them

```
$ sed -n '/ing/p' ↩
```

sed—stream editor

sed examples

- Print lines containing $100:

```
$ sed -n '/\$100/p' foo.txt
```

- Print file 10th line to the last:

```
$ sed -n '10,$p' ↩
```

- Print lines with ing's in them

```
$ sed -n '/ing/p' ↩
```

- Print just the line number of a match:

```
$ sed -n '/funny/=' foo.txt ↩
3
```

sed—stream editor

sed examples

- Print the line and its number:

```
$ sed -n -e '/fun/p' -e '/fun/=' foo.txt ↩
That was funny
3
```

sed—stream editor

sed examples

- Print the line and its number:

```
$ sed -n -e '/fun/p' -e '/fun/=' foo.txt ↩
That was funny
3
```

- Appending text:

```
$ cat my.script
/funny/p
/funny/a\
ha ha ha
$ sed -n -f my.script foo.txt ↩
That was funny
ha ha ha
```

sed—stream editor

sed examples

- Substitution:

```
$ sed -n 's/this/that/' foo.file
```

sed—stream editor

Write a sed script

- Create a script find_the.sh that will print (append) “Got One!” every time it sees the word “the” in a file.

```
#!/bin/sed -f
/the/ a\
Got one!
```

- Make the script executable:

```
$ chmod u+x find_the.sh ↩
```

sed—stream editor

Run a sed script

```
$ ./find_the.sh  owl.pussy.cat.poem ↩

"Dear Pig, are you willing to sell
                          for one shilling
Your ring?" Said the Piggy, "I will."
Got one!
So they took it away, and were married
                                  next day
Got one!
By the Turkey who lives on the hill.
Got one!
```

The Owl and the Pussy-Cat

The Owl and the Pussy-Cat went to sea
In a beautiful pea-green boat:
They took some honey, and plenty of money
Wrapped up in a five-pound note.
The Owl looked up to the stars above,
And sang to a small guitar,
”O lovely Pussy, O Pussy, my love,
What a beautiful Pussy you are,
You are,
You are!
What a beautiful Pussy you are!”

The Owl and the Pussy-Cat

Pussy said to the Owl, ”You elegant fowl,
How charmingly sweet you sing!
Oh! let us be married; too long we have tarried:
But what shall we do for a ring?”
They sailed away, for a year and a day,
To the land where the bong-tree grows;
And there in a wood a Piggy-wig stood,
With a ring at the end of his nose,
His nose,
His nose,
With a ring at the end of his nose.

The Owl and the Pussy-Cat

”Dear Pig, are you willing to sell for one shilling
Your ring?” Said the Piggy, ”I will.”
So they took it away, and were married next day
By the Turkey who lives on the hill.
They dined on mince and slices of quince,
Which they ate with a runcible spoon;
And hand in hand on the edge of the sand
They danced by the light of the moon,
The moon,
The moon,
They danced by the light of the moon.

Edward Lear

sed—stream editor

Inserting test with sed

- Create a script find_the.sh that will print (append) "Got One!" every time it sees the word "the" in a file.

```
#!/bin/sed -f
/Owl/ i\
Owl coming up!
```

- Run the script:

```
$ chmod u+x owl.sh ↩

$ ./owl.sh  owl.pussy.cat.poem ↩
Wrapped up in a five-pound note.
Owl coming up!
The Owl looked up to the stars above,
```

sed—stream editor

Answers to Questions

sed—stream editor

Answers to Questions

sed—stream editor

Answers to Questions

sed—stream editor

Answers to Questions

sed—stream editor

Answers to Questions

sed—stream editor

The End