

Searching Using Regular Expressions

Information

These notes were originally written in the year 2000 as part of a set of LPI Exam 101 training materials. The LPI training course at Bromley College was subsequently discontinued and some of the sections of the notes modified and incorporated into our one-day System Administration Courses. The remainder of the notes have now been made publicly available on the linuxtraining.org.uk website.

If you are a beginner please do not be put off of training courses by these notes, as they are rather technical. On the other hand if you are a more experienced Linux user we hope you find the coverage of this topic refreshingly clear.

For full details of our current Linux training please visit the site:

<http://ce.bromley.ac.uk/linux>

If you have reached this page from a search engine and wish to see the full contents list for the published notes please visit the site:

<http://www.linuxtraining.org.uk>

We hope you find these notes useful, but please remember that they apply to the 2.2 kernel. I will update them when I have the time.

Clive Gould - 21st December 2004

Searching Using Regular Expressions

Objective 7

Perform Searches of Text Files Making Use of Regular Expressions:
Includes creating simple regular expressions and using related tools such as grep and sed to perform searches.

Regular Expressions and Search Commands

A selection of regular expressions used with search commands such as `grep` and `sed` is listed below:

<i>Expression</i>	<i>Explanation</i>
<i>string</i>	Finds all lines containing an exact match to the <i>string</i> specified. Case sensitive by default.
<i>[Ss]tring</i>	Finds all lines containing a match to <i>String</i> or <i>string</i>
<i>[0-9]</i>	Finds all lines containing a number in the range 0 to 9
<i>[A-D]</i>	Finds all lines containing a letter in the range from A to D
<i>^String</i>	Finds all lines starting with <i>String</i>
<i>string\$</i>	Finds all lines ending with <i>string</i>
<i>'very long string'</i>	Finds all lines containing an exact match to the phrase <i>very long string</i> .

The above expressions are supported by both `grep` and `sed`, although with `sed` it is necessary to enclose them between a pair of forward slashes.

The `grep` command with the `-E` switch (cf `egrep`) supports a range extended expressions which are not listed above, but can be found on the appropriate info page.

Searching Using Regular Expressions

Global Regular Expression Parse - grep

The command `grep` searches the named input files, or standard input if no files are named, for lines containing a match to the given pattern. By default, `grep` prints the matching lines.

The syntax for `grep` is shown below:

```
grep option(s) pattern file(s)
```

Common `grep` options are:

<i>Option</i>	<i>Explanation</i>
-c	Suppress normal output and instead print a count of the number of lines which match the specified pattern.
-E	Interpret pattern as an extended regular expression. This is similar to <code>egrep</code> .
-F	Interpret pattern as a list of fixed strings, separated by newlines, any of which is to be matched. Same as <code>fgrep</code> .
-G	Interpret pattern as a basic regular expression. This is the default.
-h	Suppress listing of file name before results
-i	Ignore case in matching search patterns
-l	Return just the name of the file containing the pattern
-s	Suppress error messages for non readable files
-v	Shows all line which do not contain the pattern
-w	Searches for the pattern as a complete word, not just part of a word.

The `grep` command is very useful with log files as illustrated below:

```
[root@ext7144 log]# grep clive secure  
Nov 16 16:37:14 ext7144 login: LOGIN ON tty1 BY clive  
Nov 16 17:09:40 ext7144 login: LOGIN ON tty1 BY clive  
Nov 25 10:21:31 ext7144 login: LOGIN ON tty1 BY clive
```

Searching Using Regular Expressions

Each of the lines containing **clive** in the **secure** log file is displayed. Another example is illustrated below:

```
[clive@redhat clive]$ grep -F looked test
I looked out over the common and could see the green trees.
[clive@redhat clive]$ grep -F look* test
[clive@redhat clive]$ grep look* test
I looked out over the common and could see the green trees.
```

In the middle example above `grep` with the `-F` option has been unable to find any matches to the string `look*`, because `*` is interpreted as just another ASCII character. In the last example a match has been found as `grep` without the `-F` option has interpreted the `*` as a regular expression.

A Stream Editor - sed

We have already seen how the `sed` utility can be used for search and replace actions. The utility can also be used to search a file for a pattern and print the lines containing the pattern to standard output. By default the matches produced by `sed` are case sensitive.

Examples of using `sed` in search operations are illustrated below:

```
[clive@redhat clive]$ sed -n /looked/p test
I looked out over the common and could see the green trees.
```

```
[clive@redhat clive]$ sed -n /'The car'/p test
The car park was still wet from the heavy shower of rain.
```

```
[clive@redhat clive]$ sed -n /[0-2]/p test5
1 I looked out over the common and could see the green trees.
2 The car park was still wet from the heavy shower of rain.
```

```
[clive@redhat clive]$ sed -n /rain.$/p test
The car park was still wet from the heavy shower of rain.
```

```
[clive@redhat clive]$ sed -n '2,3 p' test
The car park was still wet from the heavy shower of rain.
I could just see the clock on the tower of St Luke's church.
```

In the last example, no pattern matching has taken place, and `sed` has been used to display lines 2 and 3 of the file `test`.