Regular Expressions (REGEXPs)

28/01/2014

Unix Regular Expressions

grep is the “find” string command in Unix. It’s used to find a string which can be specified by a REGEXP.

Regular Expression Metacharacters

. (dot)	Any one character
[...]	Any one of the characters within the square brackets
[ˆ...]	Any one of the characters not within the square brackets
ˆ	Start of line
$	End of line
\ <	Start of word
/ >	End of word
| (vertical bar)	Separates two expressions, matches either
?	Previous character (or group) is optional
+	One or more of the previous character (or group)
*	Any number (including none) of the previous character (or group) NOTE: Matches as many as possible
()	Three uses: 1: Used to enclose a pair of expressions, separated by | (vertical bar - see above) 2: Grouping for quantifiers (’?’, ’+’, and ’*’ - see above) 3: Carry some text that matches the expression within (see ’\1’, etc, below)
\1 (and \2, \3, etc)	Output the text ’carried forward’ by the brackets (see ’( )’ above).

Small RE examples

Regular Expression	Matches
bat	bat
b.t	bat, bit, b#t
bṫ	b.t
b[aeiou]t	bat, bet, bit, bot, but
bi*t	bt, bit, biit, biiiiit
ba{4}t	baaaat
ba{2,4}t	baat, baaat, baaaat
a.*z	az, a43eru, a;R*!f45

Exercise for Interesting RE examples

Create a file with following content:

```
Hi Grace!
0123456789
007 James Bond
420Thief
10240
204800
hi grace
hi GrAce
001101
The sun shines
It shines on a sunny day
evening
adam
vera
15.12.141.121
255.255.255
255.255.255.255
256.125.124.124
```

Now grep(Unix) in the file using option for extended regexp with following regular expressions and confirm the assertions:

Regular Expression	Assertions for Matches
$\hat{}[\hat{}0]*(0[\hat{}0]*)\{x\}[\hat{}0]*\$$	matches exactly x occurances of 0.
$\hat{}[\hat{}0]*(0[\hat{}0]*)\{,x\}[\hat{}0]*\$$	matches atmost x occurances of 0.
$\hat{}[\hat{}0]*(0[\hat{}0]*)\{x,\}[\hat{}0]*\$$	matches atleast x occurances of 0.
$\hat{}[\hat{}0]*(0[\hat{}0]*)\{x,y\}[\hat{}0]*\$$	matches atleast x and atmost y occurances of 0.
$\hat{}[A-Z]$	matches line starting with Capital letters.
$\hat{}[0-1]*\$$	matches lines in binary
$\hat{}[a-zA-Z[:space:][:punct:]]*\$$	matches strings with spaces and punctuation marks.
$\hat{}[0-9]*\$$	matches digits.
“$\backslash bsun\backslash b$ ”	matches the word sun.
“$\backslash bsun$”	matches the word with sun as prefix.
“$\backslash([aeiou]\backslash).\backslash 1$”	matches vowel followed by a character followed by same vowel again.
“$\backslash b((25[0-5]\vert 2[0-4][0-9]\vert[01]?[0-9][0-9]?)(\backslash.\vert\$))\{4\}\backslash b$”	matches valid IP4 addresses.