Expresiones regulares

Alberto Molina Coballes

IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

7 de noviembre de 2011

Introducción

- Las expresiones regulares son notaciones abreviadas para expresar algún criterio
- Se usan no sólo en los entornos UNIX (grep, sed, awk, . . .), también son habitales en muchos lenguajes de programación y aplicaciones (perl, python, ruby, bases de datos, XML, etc.)
- Tienes que aprender regex, son mucho más sencillas de lo que parecen ;)
- Aquí veremos las que se conocen como expresiones regulares POSIX

BRE/ERE

- Las regex se componen de caracteres normales y caracteres especiales o metacaracteres
- BRE o *Basic Regular Expressions* es la descripción inicial POSIX de regex
- ERE o *Extended Regular Expressions* añade o modifica el significado de algunos metacaracteres de BRE.
- Es importante saber si la aplicación o lenguaje de programación que estamos utilizando entiende BRE o ERE.

Metacaracteres en BRE/ERE

Carácter	BRE	ERE	Significado
\	✓	✓	Escapa el siguiente carácter
.	✓	✓	Selecciona cualquier carácter, pero sólo uno
*	✓	✓	Selecciona cualquier carácter (ninguno, uno o varios)
^	✓	✓	Selecciona la línea que empieza por . . .
$	✓	✓	Selecciona la línea que termina por . . .
[. . .]	✓	✓	Selecciona alguno de los caracteres entre los corchetes
\{n,m\}	✓	✗	Selecciona lo anterior entre n y m veces
\(\)	✓	✗	Almacena el patrón seleccionado en la posición enésima
\n	✓	✗	Utiliza el enésimo patrón almacenado
{n,m}	✗	✓	Selecciona lo anterior entre n y m veces
+	✗	✓	Selecciona una o más veces lo anterior
?	✗	✓	Selecciona una o ninguna vez lo anterior
|	✗	✓	Selecciona lo anterior o lo posterior
( )	✗	✓	Utilizado para agrupar

Ejemplos

Supongamos el fichero pepea.txt con el contenido:

```
pepe ##
pepa #
pape
#pepe
# pepe
  # pepe
```

Comprueba el resultado de ejecutar:

```
$ grep pepe pepea.txt
$ grep pep[ae] pepea.txt
$ grep ^# pepea.txt
$ grep p.p. pepea.txt
$ grep -E ^\ +#\ * pepea.txt
$ grep -E ^"(p[ea]{2})" pepea.txt # Las comillas son peculiares de grep
```

Clases de caracteres POSIX

El patrón [a-z] sólo sirve para seleccionar caracteres alfabéticos ASCII (ingleses), pero no es útil para el resto de alfabetos. Para estos casos y otros se describen en POSIX las clases de caracteres:

[:alnum:]	Caracteres alfanuméricos	[:lower:]	minúsculas
[:alpha:]	Caracteres alfabéticos	[:print:]	Caracteres imprimibles
[:blank:]	Espacio y tabulador	[:punct:]	Caracteres de puntuación
[:cntrl:]	Caracteres de control	[:space:]	Espacios en blanco
[:digit:]	Dígitos	[:upper:]	Mayúsculas
[:graph:]	Caracteres diferentes al espacio	[:xdigit:]	Dígitos hexadecimales

Ampliación de regex

- La base de las las expresiones regulares es la definición POSIX, pero hay muchas extensiones en diferentes aplicaciones y lenguajes de programación.
- Es relativamente habitual (incluso dentro de aplicaciones básicas de UNIX) encontrar expresiones regulares que no siguen BRE/ERE, por ejemplo: \w \W \< \> \b \B ...
- Conclusión: Las regex son muy útiles, pero lo son más con un buen manual al lado :)