Curso de Formação LPIC-1

Exame 101




Bruno Santos <brunosantos@whatgeek.com.pt>

Curso Linux: formação

- Expressões Regulares (ER)

Expressões Regulares

Expressões Regulares

- Basicamente, um padrão que descreve uma determinada quantidade de texto
- Nome vem da teoria matemática onde são baseadas
- Uma correspondência é um pedaço de texto, sequência de bytes ou caracteres que o motor da ER encontrou baseada no padraõ

regex – ER básica. Corresponde exactamente à palavra
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b – ER mais avançada. Descreve um endereço de email

Expressões Regulares

Expressões Regulares: Motores

- Um motor de ER é um software que processa ER
- Tenta corresponder o padrão de pesquisa a uma string data
- Não se acede ao motor directamente. A aplicação usada invoca-o quando necessário
- Diferentes motores não são totalmente compatíveis entre si

regex – ER básica. Corresponde exactamente à palavra
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b – ER mais avançada. Descreve um endereço de email

Expressões Regulares

Expressões Regulares: Estrutura

- Ancoras – Especificam a posição do padrão de procura relativamente à linha de texto
- Caracteres – Correspondêm a um ou mais caracteres numa unica posição
- Modificadores – Especificam quantas vezes o caracter anterior é repetido
- Tipos:
  - Básica
  - Extendida

Expressões Regulares

Expressões Regulares: Sintaxe

Caracteres		
Qualquer um, excepto [\^$.|?*+()	Todos os caracteres excepto os listados correspondem a eles mesmos. { e } são caracteres literais, excepto se forem parte de uma ER válida (quantificador {n})	A corresponde a A
\ (contra-barra) seguida por qualquer um dos: [\^$.|?*+(){}	Uma contra-barra suprime o significado especial dos caracteres listados	\+ corresponde a +
\Q...\E	Corresponde literalmente a qualquer caracter entre \Q e \E, suprimindo o seu significado especial	\Q+-*/\E corresponde a +-*/
\xFF onde FF são dois dígitos hexadecimais	Corresponde ao caracter com o código ASCII/ANSI FF, que depende da codificação usada.	\xA9 corresponde a © quando usado o código de página latin-1
\cA até \cZ	Correspondem ao caracter ASCII de Control+A ate Control+Z, equivalente a \x01 até \x1A. Podem ser usados em classes de caracteres	\cM\cJ corresponde a DOS/Windows CRLF + quebra de linha
\cA até \cZ	Correspondem ao caracter ASCII de Control+A ate Control+Z, equivalente a \x01 até \x1A. Podem ser usados em classes de caracteres	\cM\cJ corresponde a DOS/Windows CRLF + quebra de linha

Expressões Regulares

Expressões Regulares: Sintaxe

Classes de caracteres

Caracter	Descrição	Exemplo
[(parêntesis recto à esquerda)	Começa uma classe de caracteres. Corresponde a um caracter unico ou a todas as possibilidades oferecidas pela classe. Dentro da classe, várias regras aplicam-se.	
Qualquer caracter excepto ^-] adiciona o caracter às correspondências possíveis da classe.	Todos os caracteres excepto os caracteres especiais listados	[abc] corresponde a ou b ou c
\ (contra-barra) seguida de um qualquer ^-]	Uma contra-barra suprime o significado especial dos caracteres	[^\]] corresponde a ^ ou]
- (hífen) excepto imediatamente a seguir ao [	Corresponde a um conjunto de caracteres. (especifica um hífen se colocado imediatamente a seguir ao [])	[a-zA-Z0-9] corresponde a qualquer letra ou número
^ (acento circunflexo) imediatamente a seguir ao [	Nega a classe, fazendo corresponder a qualquer caracter não listado (especifica um acento circunflexo se colocado em qualquer lugar excepto no indicado)	[^a-d] especifica qualquer caracter excepto a, b, c ou d)
\d, \w e \s	Atalhos para digitos, palavras e espaços em branco. Podem ser usados dentro e fora das classes	
\D, \W e \S	Negação das versões em cima. Devem ser usados fora das classes (podem ser usados dentro, mas torna-se confuso)	\D corresponde a um caracter que não é um digito
[\b]	Dentro de uma classe, \b é um *backspace*	[\b\t] corresponde a um *backspace* ou TAB

Expressões Regulares

Expressões Regulares: Sintaxe

Ponto

Caracter	Descrição	Exemplo
. (ponto)	Corresponde a qualquer caracter unico, excepto quebras de linha \r e \n. Algumas ER têm uma opção para fazer o ponto corresponder a quebras de linha.	. corresponde x ou (quase) qualquer outro caracter

Ancoras

^ (acento circunflexo)	Corresponde ao inicio de linha. Corresponde a uma posição	^. corresponde a em abc\ndef. Corresponde a d em modo multi-linha
$ (dolar)	Corresponde ao final de linha. Corresponde a uma posição. Corresponde também antes da ultima quebra de linha se a string termina com uma quebra de linha.	.$ corresponde f em abc\ndef. Corresponde a c em modo multi-linha
\A	Corresponde ao inicio de uma string se a ER é aplicada a. Corresponde a uma posição. Nunca devolve resultados após uma quebra de linha	\A. corresponde a em abc
\Z	Corresponde ao final da string se a ER é aplicada a. Corresponde a uma posição. Nunca devolve resultados antes de uma quebra de linha, excepto se a ultima linha termina com uma quebra.	.\Z corresponde a f em abc\ndef
\z	Corresponde ao final de uma string se a ER é aplicada a. Corresponde a uma posição. Nunca devolve resultados após quebras de linha.	.\z corresponde a f em abc\ndef

Expressões Regulares

Expressões Regulares: Sintaxe

Palavras		
Caracter	**Descrição**	**Exemplo**
\b \B	Corresponde à posição entre um caracter imprimível (qualquer coisa que \w corresponda) e um caracter não imprimível (qualquer coisa que corresponda [^\w] ou \W) bem como ao inicio/fim de uma string se o primeiro e/ou ultimo caracter da string são caracteres de palavras	.\b corresponde c em abc
Alternações		
| (pipe)	Causa o motor de ER corresponder ou à parte esquerda ou à parte direita da expressão. Pode ser agrupado.	abc|def|xyz corresponde a abc, def ou xyz
| (pipe)	Tem a precedência mais baixa de todos os operadores. Usar o agrupamento para apenas alternar partes da expressão regular	abc(def|xyz) corresponde a abcdef ou abcxyz

Expressões Regulares

Expressões Regulares: Sintaxe

Quantificadores

Caracter	Descrição	Exemplo
? (ponto de interrogação)	Torna o item precedente opcional. Ganancioso, por isso o item apenas é incluído na correspondência se possivel.	abc? Corresponde a ab ou abc
??	Torna o item precedente opcional. preguiçoso, o item é excluído da correspondência se possível. Esta construção é às vezes excluída da documentação pelo seu uso limitado	abc?? corresponde a ab ou abc
* (asterisco)	Repete o item precedente zero ou mais vezes. Ganancioso, por isso muitos itens serão marcados antes de tentar permutações com menos correspondências do item anterior, até ao ponto onde o item precedente não é correspondido.	“.*” corresponde “def” “ghi” em abc “def” “ghi” jkl
*? (asterisco preguiçoso)	Repete o item anterior zero ou mais vezes. Preguiçoso, por isso o motor tenta falhar o item anterior antes de tentar permutações com aumento de correspondências do item anterior	“.*?” corresponde “def” em abc “def” “ghi” jkl
+ (mais)	Repete o item anterior uma ou mais vezes. Ganancioso, por isso corresponde todos os item que puder antes de tentar permutações com menos correspondências do item anterior, até ao ponto onde o item precedente é correspondido apenas uma vez	“.+” corresponde “def” “ghi” em abc “def” “ghi” jkl

Expressões Regulares

Expressões Regulares: Sintaxe

Quantificadores (continuação)

Caracter	Descrição	Exemplo
+? (mais preguiçoso)	Repete o item anterior uma ou mais vezes. Preguiçoso, pois o motor tenta corresponder o item apenas uma vez, antes de tentar permutações com aumento de correspondências do item precedente	".+?" corresponde "def" em abc "def" "ghi" jkl
{n] onde n é um inteiro >=1	Repete o item anterior exactamente n vezes	a{3} corresponde aaa
{n,m} onde n >=0 e m >=n	Repete o item anterior entre n e m vezes. Ganancioso, por isso corresponder m vezes é tentado antes de descer o número para n vezes	a{2,4} corresponde aa, aaa ou aaaa
{n,m}? onde n >=0 e m >=n	Repete o item anterior entre n e m vezes. Preguiçoso, por isso corresponder n vezes é tentado antes de subir o número para m vezes	a{2,4}? Corresponde aa, aaa ou aaaa
{n,} onde n >=0	Repete o item anterior pelo menos n vezes. Ganancioso, por isso tenta corresponder o maior número de item antes de tentar permutações com menos correspondencias, até ao ponto onde o item é correspondido apenas n vezes	a{2,} corresponde aaaaa em aaaaa
{n,}? onde n >=0	Repete o item anterior n ou mais vezes. Preguiçoso, por isso o motor corresponde primeiro o item n vezes antes de tentar permutações aumentando as correspondências.	a{2,}? Corresponde aa em aaaaa

Expressões Regulares

Expressões Regulares: POSIX

POSIX	
Valor	**Descrição**
[:digit:]	Apenas digitos de 0 a 9
[:alnum:]	Qualquer caracter alfa-numérico
[:alpha:]	Qualquer caracter alfabético
[:blank:]	Apenas espaço e TAB
[:xdigit:]	Notação hexadecimal
[:punct:]	Simbolos de pontuação . , " ' ? ! ; : # $ % & ( ) * + - / < > = @ [ ] \ ^ _ { } | ~
[:print:]	Caracteres imprimiveis
[:space:]	Qualquer caracter não imprimivel (espaço, TAB, NL, FF, VT, CR). Abreviado como \s
[:graph:]	Exclui espaços em branco (espaço, TAB). Abreviado como \W
[:upper:]	Caracteres alfabético maiusculos
[:lower:]	Caracteres alfabético minusculos
[:cntrl:]	Caracteres de controlo NL CR LF TAB VT FF NUL SOH STX EXT EOT ENQ ACK SO SI DLE DC1 DC2 DC3 DC4 NAK SYN ETB CAN EM SUB ESC IS1 IS2 IS3 IS4 DEL.

Expressões Regulares

Expressões Regulares: grep

- Ferramenta originária do mundo de Unix durante os anos 70.
- Procura entre ficheiros e directorias e vê que linhas correspondem a um determinado padrão de procura
- grep é orientado à linha. Apenas aplica a ER a cada linha do ficheiro e mostra cada linha que corresponda.
- ER não podem ser aplicadas a várias linhas.

Expressões Regulares

Expressões Regulares: sed

- Ferramenta poderosa
- Nem todos os sed's são iguais
  - Linux usa o GNU sed
  - BSD usa um próprio, diferente do GNU
- sed é orientado à linha
- Por defeito, BRE (basic regular expressions) são usadas
- Parâmetro -r para ERE (extended regular expressions)

Curso Linux

bibliografia

- LPIC I, Exam Cram 2, Brunson - QUE Certification
- LPI Linux Certification In a Nutshell, Pritchard, Pessanha, Langfeldt, Stranger & Dean – O REILLY
- Linux Administration Handbook, Second edition, Nemeth Snyder Hein – Prentice Hall
- Regular-Expressions.info - http://www.regular-expressions.info/
- IBM developerWorks – UNIX tips and tricks for the new user, Part 3: Introducing filters and regular expressions – Tim McIntire