Chega de dores de cabeça para criar expressões regulares

O Feiticeiro do Shell

Para os geeks, expressões regulares podem ser um interessante exercício mental, mas a selva de abreviações e símbolos arcanos pode ser aterrorizante para novatos. Entre no mundo do editor *txt2regex*. POR ANDREAS KNEIB

Tentar encontrar a expressão regular certa para ter como retorno a cadeia de caracteres correta é uma experiência familiarmente frustrante para a maioria dos leitores. Assim também é o sentimento de resignação que nos acomete ao descobrir que as **regex** funcionam na linguagem Perl, mas não no código Lisp no qual, por um acaso, você vinha trabalhando.

Entremos no assistente do regex, o *txt2regex*, que exploraremos neste artigo. O assistente processa padrões de texto para diversos programas e linguagens, desde *awk*, passando por *Emacs*, até *Perl*, *procmail*, *sed* e *vim*. O programa compreende principalmente um script em shell que exige o Bash versão 2.04 ou posterior.

Embora quase todas as distribuições Linux atuais possam satisfazer essa exigência, você pode querer fazer um teste para certificar-se de que está do lado do bem. Digite *echo $BASH_VERSION* ou *bash -version* para fazê-lo. Se o número de sua versão for grande o bastante, você pode continuar a instalação.

Se sua distribuição não incluir o assistente, vá à página de download do projeto em [1] para baixar um arquivo tar. Usuários do Debian podem simplesmente digitar *apt-get install txt2regex* para instalar a ferramenta.

Em seguida, descompacte o arquivo e vá para o diretório criado digitando:

```
~ > tar xvzf txt2regex-0.7.tgz
~ > cd txt2regex-0.7
```

Uma vez que não é necessário compilar o programa, você pode simplesmente digitar: *make install* como *root*:

```
~ > su
Password:senha
root# make ➥
install
```

Porém, há uma desvantagem nesse método, pois ele guarda os componentes do programa nos diretórios */usr/bin* e */usr/share/locale*, onde nada de estranho à sua distribuição deve ser instalado. Para modificar os diretórios-alvo, passe as variáveis *BINDIR* e *LOCALEDIR* para *make install*. Ver Figura 1.

Isso coloca os componentes do programa em */usr/local/bin* e */usr/local/share/locale*. Há outro método possível, que é editar as variáveis no Makefile *txt2regex-0.7/Makefile*. A variável *MANDIR* não é usada na versão 0.7 do Makefile. Você pode copiar, por exemplo, *txt2regex-0.7/txt2regex.man* para */usr/local/man/man1/*, para conseguir chamar as páginas de manual no futuro (*man txt2regex*):

```
root# cp txt2regex.man /usr/➥
local/man/man1/txt2regex.1
```

Nosso programa já está pronto para rodar. Digite *txt2regex* num terminal para iniciá-lo.

Opções

Primeiramente, vamos dar uma olhada nas opções de linha de comando da ferramenta. Há uma configuração básica para o fundo do console virtual em modo texto ou Xterm. A saída do assistente será mais facilmente legível se

GLOSSÁRIO

Regex: *Abreviação para expressão regular; uma combinação de caracteres especiais que os programas podem usar para buscar trechos específicos de texto.*

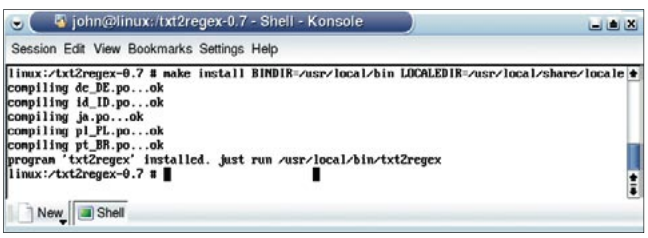

Figura 1: Passando alvos enquanto compila.

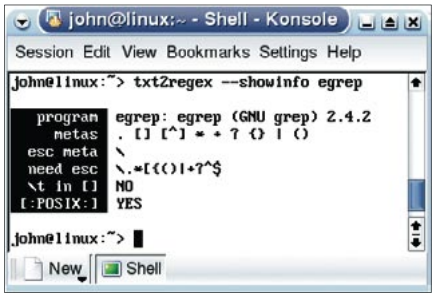

Figura 2: Exibição de meta-caracteres.

você optar por um fundo claro; o parâmetro *--whitebg* cuida disso:

```
~ > txt2regex --whitebg
```

É possível remover a cor do txt2regex especificando *--nocolor*. A opção *--prog* também é necessária para especificar os programas para os quais o script exibirá o regex. O exemplo seguinte manda a ferramenta exibir as expressões regulares para os editores vim e emacs e as linguagens de programação Visual Basic Script e Perl:

```
~ > txt2regex --prog vim,emacs,➥
 vbscript,perl
```

Para exibir todas as variações possíveis do regex, escolha a opção *--all*. Note que ao fechar o programa surge uma mensagem de erro, alertando sobre sua tela não ter linhas suficientes, mesmo se você estiver usando um Xterm grande. Para evitar que isso aconteça, você deve digitar o comando *export COLUMNS LINES* no Bash antes de iniciar o *txt2regex -all*.

Outro parâmetro importante é *--history*. Ela carrega uma expressão anteriormente definida:

```
~ > txt2regex --history ➥
'http://www.linux-magazine.com'
```

Com isso, o texto *http://www.linux-magazine.com* será exibido em várias linguagens de programação. Ao completar uma expressão, o programa exibe um atalho para a opção "history". Adicione o parâmetro *--showinfo* para obter uma lista dos meta-caracteres do programa. Veja a Figura 2.

A primeira linha contém o nome do programa, *egrep* em nosso caso, seguido por detalhes sobre os meta-caracteres, os caracteres especiais que precisam ser escapados ou as classes Posix. Para mais informações sobre o *egrep*, digite *man egrep*.

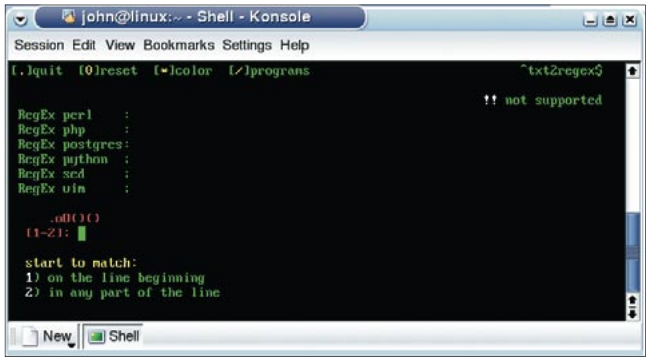

Figura 3: Txt2regex - aparência inicial.

Pergunta e Resposta

Após iniciar o assistente (ver Figura 3) digitando *txt2regex* num console, você notará as opções no topo do terminal: *quit*, *reset*, *color* e *programs*. A tecla "ponto final" fecha o editor, que exibe então uma expressão regex que pode ser acessada através da opção *--history*. Também aparece o padrão de texto especificado pelo regex, por exemplo: *Start match at beginning of line, followed by a string...*

Ao pressionar a tecla *0* para chamar a opção *reset*, o programa zera todas as entradas anteriores. Já a tecla asterisco, * , liga ou desliga a *cor*. Pressione a tecla / para acessar *Programs*, o mais importante menu de ferramentas. Ele leva a uma visão geral dos programas cujas expressões regulares você quer usar (ver Figura 4).

A seguir vem uma sessão de perguntas e respostas, ajudando você a definir sua expressão regular. O primeiro conjunto de dados, *start to match:* diz respeito ao início do padrão de busca. Pode ser *on the line beginning* (no início da linha) ou *in any part of the line* (em qualquer parte da linha). Responda pressionando *1* (para o início da linha) ou *2* (para qualquer lugar). Sua entrada será convertida para a linguagem do regex e aparece nas caixas do regex dos aplicativos especificados.

Ao terminar essa seleção, o grupo seguinte intitula-se *followed by:*. Tem uma lista de nove itens numerados, que vai de *any character* (qualquer caractere), passando por *a forbidden characters list* (uma lista de caracteres proibidos), até *anything* (qualquer coisa).

Após configurar as opções de caracteres permitidos ou proibidos, você tem que responder *quais* você quer. O editor pergunta então *quantas vezes* o caracter deve ocorrer, oferecendo sete opções que vão de *uma vez* até *até N* vezes.

Terminada essa etapa, volta-se à visão geral com o título *followed by:*. Selecionando a opção *String*, deve-se digitar o conjunto de caracteres desejado. O item *7, POSIX combination*, dá acesso a uma lista em ordem alfabética que inclui *letters* (letras), *numbers* (números), *hexadecimal numbers* (números hexadecimais) e *graphic chars* (caracteres gráficos). A tecla "ponto final" fecha o menu Posix.

O assistente usa esse método para compilar um padrão de texto para você, sem que haja necessidade de se preocupar com os aspectos específicos dos caracteres especiais, caracteres de escape ou coisas assim. Ao terminar de criar sua regex, basta pressionar a tecla "ponto final". ■

Figura 4: Seleção de linguagens de programação.

INFORMAÇÕES

[1] Página oficial do Txt2regex: *http://txt2regex.sourceforge.net/*

[2] Aurélio Marinho Jargas, autor do programa: *http://www.aurélio.net*