Linguagem de Programação: Perl

Antonio Frauzo Santos Moura
Carlos Victor Andrade de Jesus
Idyl Icaro dos Santos

Departamento de Sistemas de Informação
Universidade Federal de Sergipe (UFS) – Itabaiana, SE – Brasil

1. Introdução

Practical Extraction and Report Language, ou como é mais designada "Perl", faz menção a uma linguagem de programação concebida por volta da década de 80, mais precisamente em 1987, o grande nome nos envoltos do desenvolvimento concerne a Larry Wall. Por mais que o Perl tenha uma sintaxe "simples", o seu poder centra-se na possibilidade de articular códigos que manipulam um grande aparato de informações. Não foi à toa que a bioinformática se utiliza da mesma para o processamento de dados deveras abstrusos(todavia até têm dispõe de sentido lógico) das cadeias de ácido desoxirribonucleico.

Muitos argumentam que o processo de aprendizagem dessa linguagem seja rápido, além da sua multifuncionalidade e flexibilidade, é oportuna para manipulação de Strings, e consequentemente processamento de textos. O Perl é bem visto no mercado no ponto de vista popular, não exatamente a visibilidade, por exemplo, o seu custo benefício é elevado, pois se trata de uma linguagem gratuita, consonante a isso, é de código aberto (sucede algo como contribuição social). Sua diversificação, no que faz menção a plataformas é atraente, existe a possibilidade de ser adotada desde *Window*s, até *Linux* e *Mac* (há registro de versões para celulares). É uma linguagem simples de associar e somente valendo-se de um pouco de entendimento é exeqüível desenvolver alguns programas verdadeiramente proveitosos e claros. Vale à pena citar que o Larry teve algumas contribuições na NASA, pois bem, o Perl foi concebido justamente para simplificar as suas atividades do Laboratório de Jatos Propulsores da NASA, e como o seu trabalho ligava-se a manipulação de um significável quantitativo de dados textuais e informações complexas, o mesmo criou a linguagem de programação Perl, tornado mais fácil a observação dessas informações, e paralelo a isso, gerando relatórios (daí veio o nome em uma tradução direta," Linguagem Prática de Extração e Relatório").

Como supracitado Perl é uma linguagem de programação voltada em grande parte para manipulação de Strings, dentre as suas principais características podemos indicar: que possui suporte a orientação a objetos; é de tipagem dinâmica; foi derivada do saudoso C; no que faz alusão ao nível, seria de alto nível; é estruturada; muito freqüentemente abordada em aplicações CGI para web; multi-plataforma; e por último é gratuita.

Uma curiosidade inusitada pertinente a escolha do símbolo da linguagem, existe uma editora que atribui na capa dos seus livros acerca de linguagens de programação um animal para cada qual, aliás, a editora em questão é O'Reilly, o Larry foi questionado a respeito da escolha do camelo para a sua linguagem, o mesmo relatou que foi uma escolha deveras feliz, ele afirmou que o Perl é símile ao camelo, não é lá tão bonito e nem muito veloz, todavia unicamente com alguns goles d'água o camelo segue o seu rumo e percorre um vasto deserto.

2. Lexemas de Perl

2.1. Comentários

Para comentários de uma única linha é utilizado o símbolo `#` no início da linha do comentário.

2.2. Palavras Reservadas

__DATA__	else	lock	qw
__END__	elsif	lt	qx
__FILE__	eq	m	s
__LINE__	exp	ne	sub
__PACKAGE__		for	no
and	foreachor	unless	
cmp	ge	package	until
continue		gt	q	while
CORE	if	qq	xor	do
le	qr	y	tr	

2.3. Operadores e Delimitadores

2.3.1. Aritméticos

Operador	Operação
+	adição
-	subtração
*	multiplicação
/	divisão
%	módulo

**	expoente

2.3.2. Relacionais

Operador	Operação
==	verifica igualdade
>	verifica se é maior
<	verifica se é menor
>=	verifica se é maior ou igual
<=	verifica se é menor ou igual
!=	verifica diferença
<->	realiza comparação numérica
.	concatenação de string
=~	busca padrões em strings
!~	verifica se o padrão de string não for encontrado
=	operador de atribuição
cmp	permite diversas comparações entre strings
eq	verifica se uma string é igual a outra
ne	verifica se uma string é diferente de outra
it	verifica se uma string é inferior a outra
gt	verifica se uma string é superior a outra

le	verifica se a ordenação de uma string é inferior ou igual a outra
ge	verifica se a ordenação de uma string é superior ou igual a outra

2.3.3. Lógicos

Operador	Operação
&&	and
||	or
!	not

2.3.4. Bit a Bit

Operador	Operação
&	and
|	or
^	xor
~	inverter os bits
<<	deslocamento a esquerda
>>	deslocamento a direita

2.4 Literais String

String é uma sequência de caracteres. Na sua forma literal devem ser delimitadas por aspas simples (‘) ou duplas (“). As strings delimitadas por aspas duplas podem conter variáveis interpoladas ou caracteres especiais, usando a barra de escape.

Exemplo:

```
$variavel= “cyber”;
$aspas_simples= ‘Interacao?\n chame o $variavel.’;
$aspas_duplas= “Interacao?\n chame o $variavel.”;
print $aspas_simples;
print "\n−−−−−−−−−−−−−−−−\n";
print $aspas_duplas, "\n";
```

2.5. Literais Numérica

```
123          # inteiro 123
-123         # inteiro negativo -123
1_123        # inteiro 1234. Obs.: o underline é apenas para facilitar a leitura.
123.4        # real 123,4
.005         # real 0,005
5E-3         # real 0,005(notação científica)
23e6         # número 23.000.000
23_000_000   # número 23.000.000
23e-100      # um número muito pequeno (23 vezes 10 elevado a -100)
0xcc         # hexadecimal (valor decimal: 204) (prefixo: 0x)
0b01100100   # binário (valor decimal: 100) (prefixo: 0b)
0314 (octal) # octal (valor decimal: 204) (prefixo: 0)
```

2.6. Literal Booleano

Perl não tem um tipo booleano específico, mas todos os valores escalares- se verificado usando if, será verdadeiro ou falso. Exemplo:

```
if ( $x eq “foo” ) {
}
```

ou então,

```
if ( $x ) {
}
```

O primeiro irá verificar se o conteúdo da variável $ x é o mesmo que a string “foo”, enquanto o último irá verificar se o próprio $ x é verdadeiro ou não.

O número 0, as strings ‘0’ e ‘ ‘, a lista vazia “()” e “undef” são todos falsos em um contexto booleano. Todos os outros valores são verdadeiros.

2.7. Identificador

O identificador (nome) de uma variável pode ser constituído de letras, dígitos (0 a 9) e _ (underscore), sendo o primeiro caractere uma letra ou _, e pode ter até 252 caracteres. Estas regras também se aplicam para os demais identificadores em Perl (identificadores de funções, de rótulos, etc).

OBS: Perl é case sensitive.

2.8. Variáveis

O Perl tem 3 tipos de Variáveis: scalars, lists, e hashes.

Você usa variáveis escalar(scalars) para manipular dados escalares, como números e strings. Uma variável escalar começa com um sinal de dólar (), seguido de uma letra ou underline, depois disso, qualquer combinação de números, letras e underline. O nome de uma variável pode ser de até 255 caracteres.

Criar a variável com a palavra-chave my antes (ex: my $color = 'blue';). Isso significa que a variável ($color) tem o escopo local. Em outras palavras, a variável é local para o bloco de incisão.