Programação para Redes de Computadores 14/04/2014

Aula 09: Manipulado Strings e Funções em Bash Script

Professor: Jorge H. B. Casagrande casagrande@ifsc.edu.br
Notas de aula adaptada da original do prof. Emerson R. de Mello

.

1 Funções

O uso de funções permite o reaproveitamento de código além de tornar mais fácil a leitura do código. Uma função é uma subrotina que implementa um conjunto de operações destinada a uma tarefa específica. Trata-se de uma "caixa preta", ao invocar uma função espera-se que esta realize algumas operações, porém não é necessário saber como essas operações são executadas.

```
1  #!/bin/bash
2
3  imprimir(){
4     echo -e "\n ola mundo!"
5  }
6
7  espera(){
8     cont=1
9     while [ $cont -le 5 ]; do
10       echo -n "$cont,"
11       sleep 1
12       cont=$((cont+1))
13    done
14 }
15
16 echo "Invocando as funcoes..."
17
18 imprimir
19 espera
20 imprimir
```

Figura 1: Usando funções

1.1 Passagem de valores como argumentos

Funções podem receber argumentos e retornar o estado de saída. Tal estado indica se a função foi executada com sucesso ou não. Os argumentos passados são referenciados dentro da função com os seguintes identificadores: $1, $2, ...

```
#!/bin/bash

testandoParametros(){
 if [ -z "$1" ]
   then
      echo "O 1o. parametro e' de tamanho zero"
   else
      echo "O 1o. parametro e': $1"
 fi

 if [ "$2" ]
   then
      echo "O 2o. parametro e': $2"
 fi

 return 150
}

testandoParametros "teste"
retorno=$? # vai conter o valor retornado pela funcao invocada acima

echo "Retorno contem: $retorno"
```

Figura 2: Passando argumentos para funções

2 Cadeias de caracteres

Bash apresenta diversas operações para trabalhar com cadeias de caracteres (*strings*). Para realizar tais operações é feito uso do comando `expr` (avaliador de expressões) e também de *manipulações de variáveis*.

```
#!/bin/bash

frase="Aprendendo Bash"

# Imprimindo o conteudo da frase
echo ${frase}

# Usando o expr para imprimir o total de caracteres da frase
echo `expr length "$frase"`

# Usando expressoes regulares para imprimir o total de caracteres da frase
echo `expr "$frase" : '.*'`
```

Figura 3: Obtendo o tamanho das cadeias de caracteres

Trabalhando com subcadeias

```
#!/bin/bash

frase="Aprendendo Bash"

# Obtendo a subcadeia a partir da posicao especifica
# sintaxe: ${frase:posicao}
echo ${frase:8} # resultado: "do Bash". A 1a. posicao e' 0

# Obtendo a subcadeia de tamanho especifico a partir de uma posicao
# sintaxe: ${frase:posicao:tamanho}
echo ${frase:8:2} #resultado: "do"

# Usando indices a partir da direita
echo ${frase: -4} #resultado: "Bash". Se atente ao espaco depois dos dois pontos

# Removendo a menor subcadeia do inicio da cadeia
echo ${frase#Aprendendo} #resultado: "Bash"

# Removendo a menor subcadeia do final da cadeia
echo ${frase%Bash} #resultado: "Aprendendo"

# Substituindo a 1a. ocorrencia de uma subcadeia
echo ${frase/en/nova} #resultado: "Aprnovadendo Bash"

# Substituindo todas as ocorrencias de uma subcadeia
echo ${frase//en/nova} #resultado: "Aprnovadnovado Bash"
```

Figura 4: Trabalhando com subcadeias

```
#!/bin/bash
# renomeando todos os arquivos .TXT para .txt

antigo=TXT
novo=txt

for arquivo in $(ls *.$antigo); do
   mv -f $arquivo ${arquivo%.$antigo}.$novo
done
```

Figura 5: Exemplo: Renomeando arquivos

3 Exercícios

1. Desenvolva uma função que permita desenhar uma caixa retangular. Essa função deverá receber dois argumentos, largura e altura, para então desenhar a caixa fazendo uso de caracteres como `|`, `-` e `+`. Abaixo um exemplo de saída com os valores 10 e 5 fornecidos para largura e altura, respectivamente.

```
+---------+
|         |
|         |
|         |
+---------+
```