

Manipulando Strings

aula 12 - laboratório de SO

Manipulação de strings

Uma variável string contém uma sequência de caracteres de texto. Pode incluir letras, números, símbolos e sinais de pontuação. Alguns exemplos: **abcde**, **123**, **abcde 123**, **abcde-123**, **&acbbe=%123**

Os operadores de strings incluem aqueles que fazem **comparação**, **classificação** e encontrar o **comprimento**. A tabela a seguir demonstra o uso de alguns operadores de sequência de caracteres básicos:

Manipulação de strings

Operator	Meaning
<code>[[string1 > string2]]</code>	Compares the sorting order of <code>string1</code> and <code>string2</code> .
<code>[[string1 == string2]]</code>	Compares the characters in <code>string1</code> with the characters in <code>string2</code> .
<code>myLen1=\${#string1}</code>	Saves the length of <code>string1</code> in the variable <code>myLen1</code> .

Testando strings

```
read nome

if [ -z $nome ];then

    echo "numero vazio"

else

    echo "string digitada = $nome"

fi
```

Partes de string

Às vezes, você pode não precisar comparar ou usar uma string inteira. Para extrair os primeiros *n* caracteres de uma string podemos especificar: `${string:0:n}`. Aqui, 0 é o deslocamento na seqüência de caracteres (ou seja, o caractere para começar) onde a extração precisa ser iniciada e *n* é o número de caracteres a serem extraídos.

Para extrair todos os caracteres de uma string após um ponto (.), Use a seguinte expressão: `${string #*.}`

File Edit View Search Terminal Help

```
c7:/tmp>NAME=Eddie.Haskell
c7:/tmp>first=${NAME:0:5} ; echo first name = $first
first name = Eddie
c7:/tmp>last=${NAME#*.} ; echo last name = $last
last name = Haskell
c7:/tmp>
```

exercícios

Dica: o comando `ping -c 1 IP` - retorna um valor de pacotes caso ele encontre o IP

Faça um script que leia um numero IP e retorna se endereço responde ou não ao comando ping.

Além disso, se for digitado um ip vazio, o script deve informar ao usuario.

resposta

```
1 #!/bin/bash
2
3 echo "Entre com o ip:"
4 read numIP
5
6 #testa se a variavel esta vazia
7 if [ ! -z $numIP ]; then
8     ping -c 1 $numIP
9     if [ $? -eq 0 ]; then
10         echo "Maquina esta retornando o IP"
11     else
12         echo "Maquina nao esta Pingando"
13     fi
14 else
15     echo "IP vazio"
16 fi
17
```

exercícios

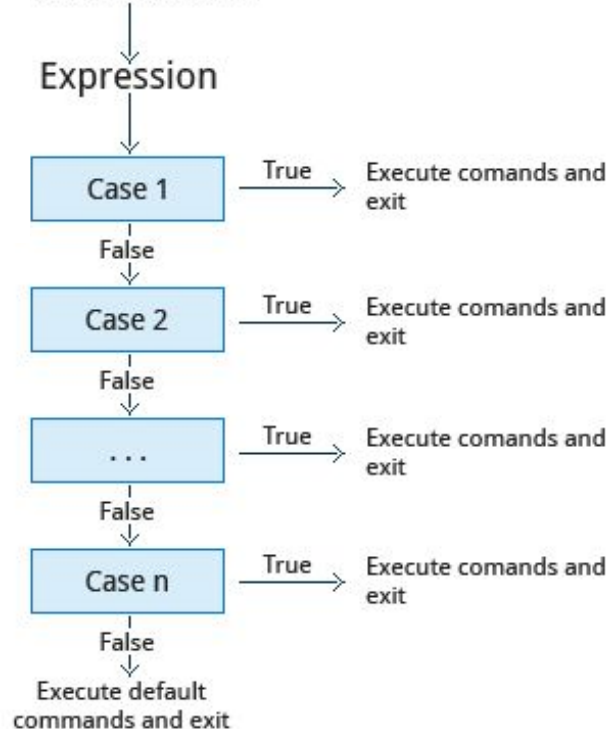
Faça um script para

- 1 - Pedir ao usuário para digitar 2 nomes
- 2 - Mostrar a string de maior tamanho
- 3 - Testar e informar se as strings são iguais ou não
- 4 - Exibir as duas strings em ordem alfabética

comando **case**

```
case expression in
  pattern1) execute commands;;
  pattern2) execute commands;;
  pattern3) execute commands;;
  pattern4) execute commands;;
  * )      execute some default commands or nothing ;;
esac
```

case Statement



Exemplo de uso do comando case:

```
# Usando o comando case
local=quarto
I
case $local in
    quarto)
        echo "Você está no quarto";;
    cozinha)
        echo "Aqui é a cozinha";;
    sala)
        echo "Está na sala agora";;
    banheiro | lavanderia)
        echo "Foi ao banheiro ou à lavanderia";;
    *)
        echo "Você não está dentro de casa";;
esac
```

exercício

```
1 #!/bin/bash
2
3 read nomearq
4 case $nomearq in
5     *.c )
6         echo "codigo C"
7         ;;
8     *.py )
9         echo "codigo python"
10        ;;
11    *.o ) ;;
12    * )
13        print "nao reconhecido."
14        return 1 ;;
15 esac
16
```