

INSTITUTO FEDERAL

Instituto Federal de Santa Catarina - IFSC
Campus São José

FIC LINUX

Introdução ao *Shell*

Prof. Francisco de Assis S. Santos, Dr.

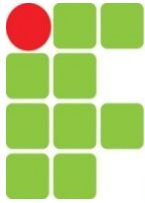
São José, 2015.

Principais comando para *Shell Script*

- **echo** - tem por objetivo imprimir mensagens no dispositivo de saída padrão, no caso o monitor.
- Abaixo algumas opções:
 - -e Ativa a interpretação de caracteres de escape (\)
 - \n -- nova linha; - n -- sem pular linha\t -- tab; \a alerta (beep)
- **read** - Permite que o usuário forneça informações via teclado (e necessário pressionar ENTER para finalizar a leitura).
- **expr** - para fazer cálculos, porém só faz operações com inteiros. Exemplo de uso:

```
soma=`expr 2 + 2` # expr entre crases
```

- **bc** - trata-se de uma calculadora, ideal para quando necessitamos efetuar cálculos com números reais. Exemplo de uso:

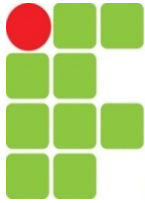


Shell Script: Variáveis

```
#!/bin/bash
# Isto é um comentário. Todo texto apos o caracter # será
ignorado
echo "Trabalhando com variáveis"
a=1
b=2
c=`expr $a + $b` # a expressão está entre crases
d=$((c+a))
echo "O valor de a é $a, o valor de b é $b, o valor de c é' $c e o
valor de d é $d"
curso="FIC Linux 2015"
echo "O conteúdo de curso e' $curso"

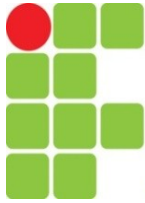
# outro exemplo
versao=$(uname -r)

echo "A versao do kernel e' $versao"
```



Shell Script: Operadores Relacionais e Lógicos

Numéricos		Cadeia de caracteres		Operadores lógicos	
-eq	Igual	=	igual	-a	E lógico (AND)
-ne	Diferente	!=	diferente	&&	E lógico (AND)
-ge	Maior ou igual	-n	Não é nula	-o	OU lógico (OR)
-le	Menor ou igual	-z	É nula		OU lógico (OR)
-gt	Maior			!	negação
-lt	menor				



Shell Script: Estruturas de Decisão

Se...então...senão

```
#!/bin/bash

nota=5

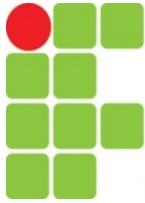
if [ $nota -ge 5 ];
then

echo "nota maior ou igual a 5"

else

echo "nota menor que 5"

fi
```



Shell Script: Estruturas de Decisão

Se...então...senão

```
#!/bin/bash

nota=5

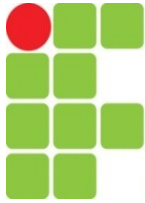
if [ $nota -ge 5 ];
then

echo "nota maior ou igual a 5"

else

echo "nota menor que 5"

fi
```



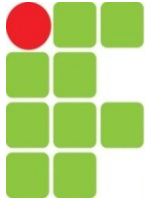
Shell Script: Estruturas de Decisão

Se...então...senão com o
operador AND

```
#!/bin/bash

a=3
b=2
c=1

# usando o operador E
if [ $a -gt $b ] && [ $a -gt $c ];
then
echo "A é o maior"
else
echo "A não é o maior"
fi
```



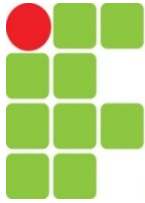
Shell Script: Estruturas de Decisão

Escolha ...caso...

```
#!/bin/bash

echo -n "Entre com um numero de 1 a 5: "
read numero

case $numero in
1)
echo "Voce escolheu 1"
;;
2)
echo "Voce escolheu 2"
;;
3)
echo "Voce escolheu 3"
;;
4 | 5)
echo "Voce escolheu 4 ou 5"
;;
*)
echo "Voce escolheu um numero diferente de 1, 2, 3, 4 ou 5"
;;
esac
```

Shell Script: Estruturas de Repetição

Enquanto

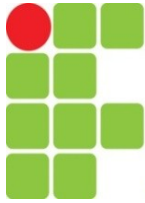
```
#!/bin/bash

num=10

while [ $num -gt 0 ]; do
echo "contando $num"
num=$((num-1))
done

#-----#
#usando o operador de negacao '!'
num=10

while ! [ $num -eq 0 ]; do
echo "contando $num"
num=$((num-1))
done
```



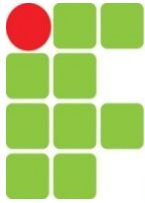
Shell Script: Estruturas de Repetição

Para (for)

```
#!/bin/bash

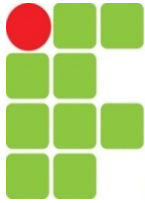
# maneira tradicional do shell
for contador in `seq 1 10`; do
    echo $contador
done

# percorrendo uma lista de palavras separadas por espaco
lista="Cabeamento Gerencia Programacao Equipamentos"
for disciplina in $lista; do
    echo "Disciplina $disciplina"
done
# Dica: que tal listar os arquivos de um diretorio?
# lista=`ls` # o ls esta' entre crases
```



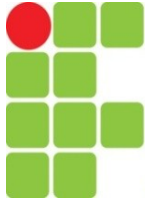
Exercícios

- 1) Desenvolva um algoritmo que leia um número inteiro positivo e imprima a sequência de 0 até este número. O programa deverá tratar caso o usuário forneça um número menor que zero.
- 2) Desenvolva um script em bash que conte quantos arquivos possui um determinado diretório.
- 3) Desenvolva um algoritmo que simule a autenticação de usuários. O usuário deve fornecer uma senha e se esta senha for igual a palavra secreta deverá exibir a mensagem "Acesso autorizado", caso contrario deverá exibir "Acesso negado". O algoritmo deverá solicitar a senha ao usuário ate que este forneça a senha correta ou ate que o número de tentativas permitidas seja alcançado. No caso, o número máximo de tentativas e 3.



Funções

O uso de funções permite o reaproveitamento de código além de tornar mais fácil a leitura do código. Uma função é uma sub-rotina 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.



Funções: Sem Passagem de Argumentos e Sem Retorno de Parâmetro

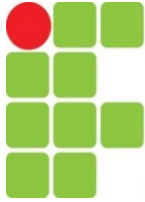
```
#!/bin/bash

imprimir(){
    echo -e "\n ola mundo!"
}

espera(){
    cont=1
    while [ $cont -le 5 ]; do
        echo -n "$cont,"
        sleep 1
        cont=$((cont+1))
    done
}

echo "Invocando as funcoes..."

imprimir
espera
imprimir
```



Funções: Com passagem de Argumentos e Retorno Parâmetro

```
#!/bin/bash

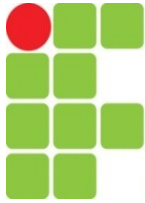
testandoParametros(){
  if [ -z "$1" ]
  then
    echo "0 1o. parametro e' de tamanho zero"
  else
    echo "0 1o. parametro e': $1"
  fi

  if [ "$2" ]
  then
    echo "0 2o. parametro e': $2"
  fi

  return 150
}

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

echo "Retorno contem: $retorno"
```



Exercício

1. Desenvolva um *script* em bash que apresente ao usuário algumas opções:

- 1 – Mostrar as configurações de Rede
- 2 – Baixar as configurações da Placa de Rede
- 3 – Carregue as configurações da Placa de Rede
- 4 – Mostrar as Portas TCP/IP abertas e que estão em uso
- 5 – Possibilitar o usuário digitar um endereço de site e obter o caminho entre a máquina local e o Servidor
- 6 – Permita o usuário obter o endereço IP de um site da web, informando o endereço do mesmo
- 7 – Sair do aplicativo

Bibliografia

MORAES. P. S. 2000. UNICAMP – Centro de Computação. Lógica de Programação. Disponível em:
<http://www.lab.ufra.edu.br/lasic/images/AULAS/PROF_CHASE/TEC_PROG_I/UFRA_TEC_PROG_P_01.pdf>. Acessado em: 08/2014.

J. L. Güntzel e F. A. Nascimento, Introdução aos Sistemas Digitais, Vol. 1, 2001.

J. H. C. Casagrande. Notas de Aula. Acessado em: Julho de 2014, Disponível em: <<http://www.sj.ifsc.edu.br/casagrande/PRC>>.