

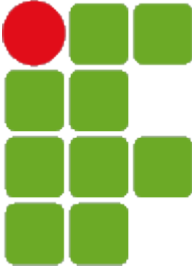
Instituto Federal de Santa Catarina  
CST em Análise e Desenvolvimento de Sistemas  
Introdução à Computação

*Shell Script*

Prof. Diego da Silva de Medeiros

São José, agosto de 2024

# @ que é

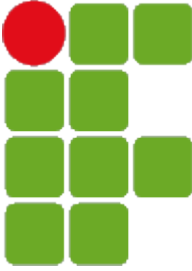


- Sequência de comandos, organizados num arquivo executável
- “Linguagem de programação”, interpretada, não compilada
- Interpretado por algum programa do tipo Shell
- Operações típicas:
  - Manipulação de arquivos
  - Instalação e execução de programas
  - Configuração automatizada do sistema
  - Execução de tarefas repetitivas



```
aluno@Maquina:~$ file /usr/bin/* | grep shell
```

# Primeiro script



- Criar um arquivo de texto chamado script.sh

```
aluno@Maquina:~$ gedit script.sh &
```

- Escrever o código abaixo, salvar e fechar o arquivo

```
#!/bin/sh  
  
clear  
  
# Imprime a mensagem "Olá mundo!" na tela  
  
echo "Olá mundo!"
```

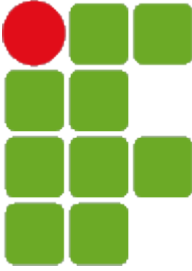
- Dar permissão de execução para o arquivo

```
aluno@Maquina:~$ chmod a+x script.sh
```

- Executar o script

```
aluno@Maquina:~$ ./script.sh
```

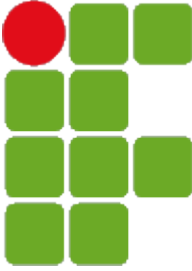
# Características do primeiro script



```
#!/bin/sh  
  
clear  
  
# Imprime a mensagem "Olá mundo!" na tela  
  
echo "Olá mundo!"
```

- **#!/bin/bash**
  - Shebang, sha-bang, hashbang, pound-bang, hash-pling
  - Especifica o endereço absoluto para o interpretador de comandos a ser usado na execução da script
- **Comentários**
  - Scripts DEVEM ser ricamente comentadas, usando o caractere “#” como início
- Script é encerrada quando execução chega ao fim do arquivo (não é necessário nenhum tipo de comando de encerramento ou retorno)
- Pode ser alterada a qualquer momento

# Variáveis



- Linguagem não tipada
- Nomes de variáveis: sequência de caracteres que deve começar por qualquer letra maiúscula ou minúscula ou underscore (\_)
- Não pode haver espaço entre o nome, o símbolo de igual, e o valor
- Valor da variável é obtido usando o cifrão "\$"
- Nome da variável é obtido usando "\\$"
- Atribuir saídas de comandos à variáveis, usa-se `variavel=$(comando)` ou `variavel=`comando``
- Variáveis são removidas com o comando `unset`

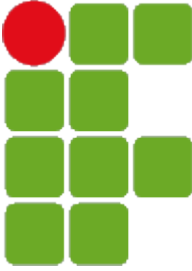
```
MENSAGEM_DATA=1979
autor="Stephen Bourne"
VAR=ola
_MENSAGEM2=oi-2.020

echo "O autor é $autor"
echo "O nome da variável é \${VAR}"

resultado=$(ls -l)
echo "O resultado do comando é: $resultado"

unset autor VAR
```

# Argumentos na chamada do comando



- Os argumentos na chamada do comando são acessados através das variáveis \$1, \$2, \$3, ..., \$9
- A quantidade total de parâmetros passada é acessada através da variável \$#

```
aluno@Maquina:~$ cat script.sh
```

```
#!/bin/bash
```

```
echo "Total de argumentos: $#"
```

```
echo "Primeiro argumento: $1"
```

```
echo "Segundo argumento: $2"
```

```
echo "Terceiro argumento: $3"
```

```
aluno@Maquina:~$ ./script.sh ola mundo
```

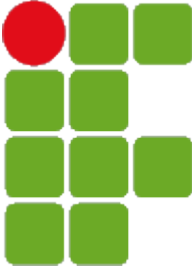
```
Total de argumentos: 2
```

```
Primeiro argumento: ola
```

```
Segundo argumento: mundo
```

```
Terceiro argumento:
```

# Variáveis globais (de ambiente)



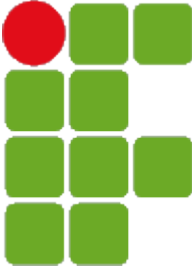
- Variáveis que podem ser utilizadas por múltiplos scripts em Shell
- Definidas com o comando `export`
- Principais:
  - `PATH`: define diretórios de procura por programas executados no shell;
  - `USER`: informa o nome do usuário do shell;
  - `HOME`: informa o caminho do diretório home do usuário;
  - `LANG`: Idioma/Linguagem, especificada como locale;
  - `PWD`: diretório atual;
  - `TERM`: Tipo de terminal atual em uso.
  - `UID`: UID do usuário atual.
  - `RANDOM`: Gera um número aleatório
- Ver variáveis de ambiente

```
aluno@Maquina:~$ env | grep -v '[:;]' | sort
```

- Definir e exportar variáveis globais

```
VARIABEL1=Teste
export VARIABEL1          # Exporta variável já existente
export VARIABEL2=Teste    # Define e exporta variável em uma única linha
```

# Arrays



- Tipo de dado que armazena um conjunto de valores
- Não compatível com Bourne Shell (sh), mas compatível com o Bourne-Again Shell (Bash)

```
aluno@Maquina:~$ cat script.sh
```

```
#!/bin/bash
```

```
array_numero=(1 22 3 4 5 6 7 8 9)
```

```
array_string=("abc" "def" "ghi")
```

```
echo "${array_numero[2]}"
```

```
echo "${array_string[1]}"
```

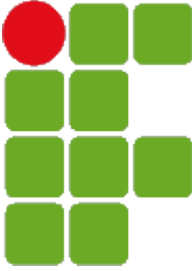
```
aluno@Maquina:~$ ./script.sh
```

```
3
```

```
def
```



# *Entrada de dados do usuário*



- Interação com o usuário
- Realizado através do comando `read variavel`

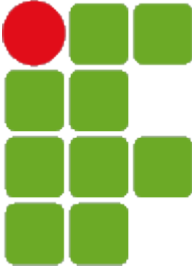
```
#!/bin/bash

echo "Qual o nome de uma de suas músicas favoritas?"

read nome_musica;

echo "Você gosta de ouvir $nome_musica!"
```

# Expressões aritméticas



- Realizado através da expressão `$ ( ( . . . ) )`

```
#!/bin/bash
```

```
echo  $ ( (2*3) )
```

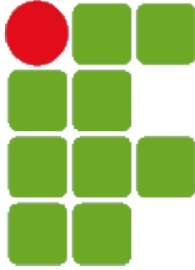
```
NUM=44
```

```
echo  $ ( (NUM*2) )
```

```
NUM=$ ( (NUM+1) )
```

```
echo  $NUM
```

# Condicional: IF, ELIF e ELSE



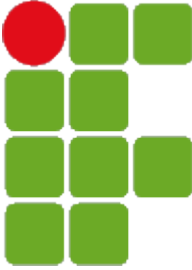
```
#!/bin/bash

echo "Selecione uma opção:"
echo "1 - Exibir data e hora do sistema"
echo "2 - Exibir o resultado da divisão 10/2"
echo "3 - Exibir uma mensagem"

read opcao;

if [ $opcao == "1" ];
then
    data=$(date +"%T, %d/%m/%y, %A")
    echo "$data"
elif [ $opcao == "2" ];
then
    result=$((10/2))
    echo "divisao de 10/2 = $result"
elif [ $opcao == "3" ];
then
    echo "Informe o seu nome:"
    read nome;
    echo "Bem-vindo ao mundo do shell script, $nome!"
else
    echo "Nenhuma opção selecionada!"
fi
```

# Condicional: CASE



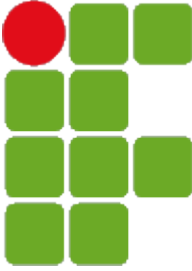
```
#!/bin/bash

echo "Selecione uma opção:"
echo "1 - Exibir data e hora do sistema"
echo "2 - Exibir o resultado da divisão 10/2"
echo "3 - Exibir uma mensagem"

read opcao;

case $opcao in
    "1")
        data=$(date +"%T, %d/%m/%y, %A")
        echo "$data"
        ;;
    "2")
        result=$((10/2))
        echo "divisao de 10/2 = $result"
        ;;
    "3")
        echo "Informe o seu nome:"
        read nome;
        echo "Bem-vindo ao mundo do shell script, $nome!"
        ;;
esac
```

# Repetição: FOR



- Vários formatos diferentes:
  - `for i in {10..0};`
  - `for i in $(seq 1 2 10)`
  - `for ((i=1; i<5; i++))`

```
#!/bin/bash

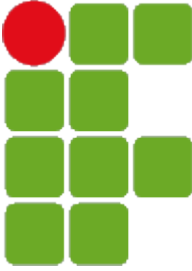
echo "Testando o loop for"

for i in {10..0};
do
    echo "$i"
done

echo "Loop FOR com palavras"

for i in um dois tres quatro cinco
do
    echo "$i"
done
```

# Repetição: WHILE



```
#!/bin/bash

echo "Menu de seleção:"
echo "1. Exibir mensagem de saudação"
echo "2. Exibir o dia e a hora"
echo "3. Sair"

while read -p "Qual opção você deseja? " entrada
do

    if [ $entrada -eq 1 ]; then
        echo "Bem-vindo(a)!"

    elif [ $entrada -eq 2 ]; then
        date

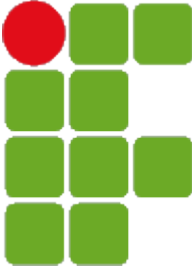
    elif [ $entrada -eq 3 ]; then
        date
        break

    else
        echo "Digite 1, 2 ou 3!"

    fi

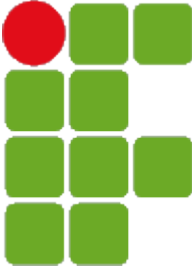
done
```

# Referências



- Muitas referências em **Aurelio.net**, em especial a apostila **Introdução ao Shell Script**
- **Shell Script: um guia básico**, de Diego Mariano
- **Introdução ao Shell Script no Linux**, do DevMedia

# *Exercícios*



- Fazer Laboratório 1 e 2 disponível na aula de hoje no SIGAA