



Introdução à Ciência da Computação – Lista 8

Shell script – parte 5

Nome: RA:

- 1) Qual variável de ambiente guarda o valor lido pelo comando read, caso não tenha nenhuma criada para guardar o valor? Faça um script utilizando essa variável de ambiente, onde a entrada será o nome do mês atual. Mostre a execução. Depois acrescente um timer de 8 segundos, como mostrado em aula e apresente a execução.

```
GNU nano 7.2 script.sh *
#!/bin/bash

echo "Enter the current month:"
read
echo "The current month is: $REPLY"

echo "Waiting for 8 seconds..."
sleep 8
echo "Done waiting."
```

```
pedro@DESKTOP-UAUJT7E:~$ nano script.sh
pedro@DESKTOP-UAUJT7E:~$ chmod +x script.sh
pedro@DESKTOP-UAUJT7E:~$ ./script.sh
Enter the current month:
junho
The current month is: junho
Waiting for 8 seconds...
Done waiting.
```

- 2) Crie um arquivo txt sobre você (nome, idade, cidade, hobbies, etc) e faça a leitura do arquivo com o comando read. Apresentando todo o conteúdo do arquivo no terminal.

```
GNU nano 7.2 eu.txt *
Nome: Pedro Ferreira
Idade: 20
Cidade: Machado MG_
```

```
GNU nano 7.2 read.sh *
#!/bin/bash

echo "Reading content from my_info.txt:"
while IFS= read -r line; do
    echo "$line"
done < eu.txt
```

```
pedro@DESKTOP-UAUJT7E:~$ nano eu.txt
pedro@DESKTOP-UAUJT7E:~$ nano read.sh
pedro@DESKTOP-UAUJT7E:~$ chmod +x read.sh
pedro@DESKTOP-UAUJT7E:~$ ./read.sh
Reading content from my_info.txt:
Nome: Pedro Ferreira
Idade: 20
Cidade: Machado MG
```

- 3) Desenvolva um script utilizando o comando case. Cada opção chama uma função e apresenta o resultado da função. Exemplos:

Digite um número entre 1 e 4: 4 Digite um número: 5 Subtraindo 10: -5	Digite um número entre 1 e 4: 1 Digite um número: 10 Multiplicando por 10: 100
---	--

4)

Modifique o script abaixo para que apresente o dobro apenas dos números entre 20 e 30.

```
GNU nano 7.2                                caso1.sh *
#!/bin/bash

# Função para subtrair 10
subtrair_dez() {
    local num=$1
    echo "Subtraindo 10: $((num - 10))"
}

# Função para multiplicar por 10
multiplicar_por_dez() {
    local num=$1
    echo "Multiplicando por 10: $((num * 10))"
}

echo "Digite um número entre 1 e 4 para escolher uma operação:"
read escolha

echo "Digite um número para a operação:"
read numero_usuario

case $escolha in
    1)
        subtrair_dez "$numero_usuario"
        ;;
    2)
        multiplicar_por_dez "$numero_usuario"
        ;;
    3)
        echo "Opção 3 selecionada (nenhuma função específica aqui para exemplo)"
        ;;
    4)
        echo "Opção 4 selecionada (nenhuma função específica aqui para exemplo)"
        ;;
    *)
        echo "Escolha inválida."
        ;;
esac
```

```
pedro@DESKTOP-UAUJT7E: ~
GNU nano 7.2                                caso2.sh *
#!/bin/bash

# Função para calcular o dobro se o número estiver entre 20 e 30
calcular_dobro_se_valido() {
    local num=$1
    if (( num >= 20 && num <= 30 )); then
        echo "O dobro de $num é: $((num * 2))"
    else
        echo "$num não está entre 20 e 30. Nenhum dobro calculado."
    fi
}

echo "Digite um número para verificar se ele pode ser dobrado (entre 20 e 30):"
read numero_entrada
calcular_dobro_se_valido "$numero_entrada"
```

5) Crie um script como whiptail, receba um número e mostre o dobro daquele número.

```
GNU nano 7.2                                whiptail.sh *
#!/bin/bash

# Verifica se o whiptail está instalado
if ! command -v whiptail &> /dev/null
then
    echo "whiptail não foi encontrado. Por favor, instale-o (ex: sudo apt-get install whiptail)."
    exit 1
fi

input=$(whiptail --inputbox "Digite um número para calcular o seu dobro:" 10 50 3>&1 1>&2 2>&3)

status_saida=$?
if [ $status_saida = 0 ]; then
    if [[ "$input" =~ ^[0-9]+$ ]]; then # Validação básica para número inteiro
        dobro=$((input * 2))
        whiptail --msgbox "O dobro de $input é: $dobro" 10 50
    else
        whiptail --msgbox "Entrada inválida. Por favor, digite um número válido." 10 50
    fi
else
    whiptail --msgbox "Você cancelou a operação." 10 50
fi
```

O dobro de 4 é: 8

<Ok>

- 6) Escolha um widget do comando dialog e mostre sua execução. Escolha um diferente dos que foram mostrados em aula.

```
GNU nano 7.2                                check.sh *
#!/bin/bash

# Verifica se o dialog está instalado
if ! command -v dialog &> /dev/null
then
    echo "dialog não foi encontrado. Por favor, instale-o (ex: sudo apt-get install dialog)."
    exit 1
fi

# Exibe uma checklist
exec 3>&1
selecao=$(dialog --checklist "Selecione suas frutas favoritas:" 20 60 5 \
"Maçã" "Uma fruta comum" ON \
"Banana" "Uma fruta amarela" OFF \
"Laranja" "Uma fruta cítrica" OFF \
"Uva" "Pequena e doce" OFF \
"Morango" "Vermelho e delicioso" ON \
2>&1 1>&3)
exec 3>&-

status_saida=$?

if [ $status_saida -eq 0 ]; then
    dialog --msgbox "Você selecionou: $selecao" 10 50
else
    dialog --msgbox "Você cancelou a seleção." 10 50
fi
```

7) Desenvolva um ckecklist, utilize sua criatividade. Pode ser whiptail ou dialog.

```
GNU nano 7.2                                compras.sh *
#!/bin/bash

# Arquivo onde os itens do checklist serão salvos
CHECKLIST_FILE="meu_checklist_simples.txt"

# Cria o arquivo se ele não existir
touch "$CHECKLIST_FILE"

echo "--- SEU CHECKLIST ---"
cat "$CHECKLIST_FILE" # Exibe o conteúdo atual do arquivo
echo "-----"

read -p "Adicionar novo item (deixe em branco para sair): " novo_item

if [ -n "$novo_item" ]; then # Verifica se o input não está vazio
    echo "[] $novo_item" >> "$CHECKLIST_FILE" # Adiciona o item com status pendente
    echo "Item '$novo_item' adicionado."
else
    echo "Nenhum item adicionado."
fi_
```