



Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: Larissa Rodrigues de Ávila

RA: 2024.1.08.031

- 1) Crie um script chamado escrevenome, faça com que a saída desse script seja seu nome completo. Não utilize o comando chmod. Depois crie um script chamado testecompara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevenome. Mostre o resultado da saída.

```
Open  [icon] *escrevenome.sh ~/ Save [menu]
1 #!/bin/bash
2
3 echo "Larissa Rodrigues de Ávila"
4
```

```
Open  [icon] testecompara.sh ~/ Save [menu]
1 #!/bin/bash
2 arquivo=/home/2024.1.08.031/escrevenome.sh
3
4 if [ -r arquivo ] && [ -x arquivo ]
5 then
6     echo "O usuário tem permissão para leitura e execução"
7 else
8     echo "O usuário logado não tem permissão de execução"
9 fi
10
```

```
2024.1.08.031@suporte-OptiPlex-3050:~$ gedit escrevenome.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ gedit testecompara.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ chmod 755 testecompara.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ ./testecompara.sh
O usuário logado não tem permissão de execução
```

- 2) Crie um script chamado `frutacase`. Com base no valor da variável `fruta` mostre uma breve descrição da fruta. Faça com 5 frutas. Exemplo: `fruta=uva`, echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, musses, geleias, tortas, gelatinas, sucos."

```
frutacase.sh
1 #!/bin/bash
2
3 fruta=goiaba
4
5 case $fruta in
6   goiaba)
7     echo "A goiaba é fruto da goiabeira, árvore da espécie Psidium guajava, da família
8     Myrtaceae, originária da América tropical. A fruta é utilizada em diferentes produtos
9     derivados, tais como goiabadas, doces, compotas, sucos, sorvetes e molhos salgados.;;
10   banana)
11     echo "A banana é uma pseudobaga da bananeira, uma planta herbácea da família
12     Musaceae, originárias do sudeste da Ásia. Existem diversas variedades de cultivo, entre
13     elas estão a banana-da-terra, a banana-nanica e a banana-maçã.;;
14   manga)
15     echo "A manga é fruto da mangueira, árvore frutífera da família Anacardiaceae,
16     nativa do sul e do sudeste asiático. Possui coloração variada: amarelo, laranja e
17     vermelha. Normalmente, quando não está madura, sua cor é verde.;;
18   melao)
19     echo "O melão é um fruto cujo nome científico é "cucumis melo" e pertence a família
20     cucurbitaceae, nativo do Oriente Médio. A abundância de água em seu interior e o sabor
21     suave tornam o melão uma fruta muito apreciada na forma de refrescos.;;
22   pera)
23     echo "A pera é o fruto da pereira, uma árvore do gênero Pyrus, pertencente a
24     família Rosaceae. É originária da Ásia, provavelmente da China, e conta com mais de 20
25     variedades de espécies cultivadas. Pode ser consumida in natura, enlatada e em suco.;;
26   *)
27     echo "A fruta não está na lista";;
28 esac
```

```
2024.1.08.031@suporte-OptiPlex-3050:~$ gedit frutacase.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ chmod 755 frutacase.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ ./frutacase.sh
A goiaba é fruto da goiabeira, árvore da espécie Psidium guajava, da família Myr
taceae, originária da América tropical. A fruta é utilizada em diferentes produt
os derivados, tais como goiabadas, doces, compotas, sucos, sorvetes e molhos sal
gados.
```

3) Cite, explique e faça um script simples para cada estrutura de repetição do shell bash. Use sua criatividade para os scripts.

- FOR: O comando for permite criar um loop que percorre uma lista de valores, e a cada iteração executa um conjunto definido de comandos para cada valor da lista.

```
Open  [icon]  for.sh  Save
1 #!/bin/bash
2
3 for i in Nayene Leandro Camilla Felipe
4 do
5     echo "Eu tenho um amigo chamado: $i"
6 done
```

```
2024.1.08.031@suporte-OptiPlex-3050:~$ gedit for.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ chmod 755 for.sh
2024.1.08.031@suporte-OptiPlex-3050:~$ ./for.sh
Eu tenho um amigo chamado: Nayene
Eu tenho um amigo chamado: Leandro
Eu tenho um amigo chamado: Camilla
Eu tenho um amigo chamado: Felipe
2024.1.08.031@suporte-OptiPlex-3050:~$
```

- WHILE: O comando while permite que uma determinada condição seja testada previamente; o que decidirá se os seguintes comandos serão ou não executados. Enquanto o teste for verdadeiro para cada variável, os comandos seguintes serão executados. Se o teste for falso, a sequência de comandos pertencentes ao while é finalizada.

```
Abrir  [icon]  while.sh
1 #!/bin/bash
2 echo "Números pares de 1 a 20"
3 var=2
4
5 while [ $var -le 20 ]
6 do
7     echo $var
8     var=$(( $var + 2 ))
9 done
10
```

```
larissa-avila@larissa-avila:~$ chmod 755 while.sh
larissa-avila@larissa-avila:~$ ./while.sh
Números pares de 1 a 20
2
4
6
8
10
12
14
16
18
20
```

- UNTIL: É o oposto do while. Enquanto o teste for falso, os comandos dentro do until serão executados, quando teste for verdadeiro, o looping para.

A screenshot of a code editor window. The title bar shows 'until.sh' and a file icon. The editor content is a shell script named 'while.sh'. The script starts with a shebang, prints a header, sets a variable, and uses a 'until' loop to print even numbers from 2 to 20. The script is 8 lines long.

```
1 #!/bin/bash
2 echo "Números pares de 1 a 20"
3 var=2
4 until [ $var -eq 22 ]
5 do
6     echo $var
7     var=$(( $var + 2 ))
8 done
```

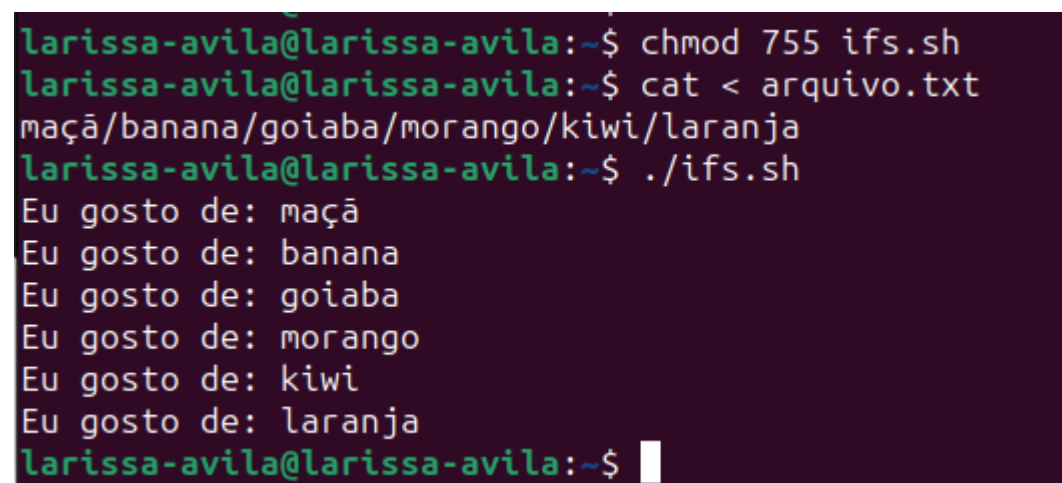
```
larissa-avila@larissa-avila:~$ chmod 755 until.sh
larissa-avila@larissa-avila:~$ ./until.sh
Números pares de 1 a 20
2
4
6
8
10
12
14
16
18
20
larissa-avila@larissa-avila:~$
```

4) Explique o que é IFS e faça um script diferente do que foi visto em aula. Use sua criatividade.

A variável IFS define uma lista de caracteres que, por padrão, o shell bash usa como separadores de campos. Essa lista pode ser alterada.



```
1 #!/bin/bash
2 arq=arquivo.txt
3 IFSOLD=$IFS
4 IFS=$"/"|
5
6 for i in `cat $arq`
7 do
8     echo "Eu gosto de: $i"
9 done
10 IFS=$IFSOLD
11
```



```
larissa-avila@larissa-avila:~$ chmod 755 ifs.sh
larissa-avila@larissa-avila:~$ cat < arquivo.txt
maçã/banana/goiaba/morango/kiwi/laranja
larissa-avila@larissa-avila:~$ ./ifs.sh
Eu gosto de: maçã
Eu gosto de: banana
Eu gosto de: goiaba
Eu gosto de: morango
Eu gosto de: kiwi
Eu gosto de: laranja
larissa-avila@larissa-avila:~$
```

5) Crie um script for no estilo C que mostre na tela os números de 50 a 20.

```
Abrir  ▾  [ícone]  forc.sh  ~/
1  #!/bin/bash
2
3  for (( i = 50; i >= 20; i-- ))
4  do
5      echo "$i"
6  done
7
```

```
larissa-avila@larissa-avila:~$ chmod 755 forc.sh
larissa-avila@larissa-avila:~$ ./forc.sh
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
```

- 6) Desenvolva um script que receba um parâmetro e verifique se o valor está entre 0 e 10. Caso sim mostre o triplo do valor. Caso ele esteja entre 10 e 20 mostre o dobro. Caso não esteja nos anteriores apresente uma mensagem.

```
1 #!/bin/bash
2
3 if [ $1 -ge 0 ] && [ $1 -le 10 ]
4 then
5     primeiro=$(( $1 * 3 ))
6     echo $primeiro
7 elif [ $1 -gt 10 ] && [ $1 -le 20 ]
8 then
9     segundo=$(( $1 * 2 ))
10    echo $segundo
11 else
12    echo "O número não está entre 0 e 20"
13 fi
```

```
larissa-avila@larissa-avila:~$
larissa-avila@larissa-avila:~$ chmod 755 parametro.sh
larissa-avila@larissa-avila:~$ ./parametro.sh 4
12
larissa-avila@larissa-avila:~$ ./parametro.sh 12
24
larissa-avila@larissa-avila:~$ ./parametro.sh 30
O número não está entre 0 e 20
```

7) Explique o que é \$# e faça um script diferente do que foi visto em aula. Faça com dois parâmetros. Use sua criatividade.

\$# é uma variável que contém o número de parâmetros de linhas de comando fornecidos ao rodar o script

Abrir ▾

ultimo.sh

~/

```
1 #!/bin/bash
2 if [ $# -ne 2 ]
3 then
4     echo "O programa só pode ser feito com dois parâmetros"
5 else
6     mult=$(( $1 * $2 ))
7     echo "$1 vezes $2 é igual a $mult"
8 fi
```

```
larissa-avila@larissa-avila:~$ chmod 755 ultimo.sh
larissa-avila@larissa-avila:~$ ./ultimo.sh 2 3
2 vezes 3 é igual a 6
larissa-avila@larissa-avila:~$ ./ultimo.sh 1
O programa só pode ser feito com dois parâmetros
larissa-avila@larissa-avila:~$ ./ultimo.sh 2 3 4
O programa só pode ser feito com dois parâmetros
larissa-avila@larissa-avila:~$ ./ultimo.sh 10 4
10 vezes 4 é igual a 40
larissa-avila@larissa-avila:~$
```