



Introdução à Ciência da Computação – Lista 4

Shell script – parte 1

Nome: Luiz Augusto Pereira Hipolito

RA: 2025.1.08.033

1) O que é Shell?

Interface entre o usuário e o sistema operacional. Ele interpreta comandos e os executa.

O que é Shell Scripting?

Criação de scripts (arquivos de texto) contendo comandos do shell para automatizar tarefas.

Dê exemplos de shell existentes no padrão Unix.

bash (Bourne Again SHell)

sh (Bourne Shell)

zsh (Z Shell)

ksh (Korn Shell)

csh (C Shell)

2) O que são variáveis de ambiente? Qual a diferença de variáveis globais e locais?

Variáveis globais: Estão disponíveis para o shell atual e processos filhos.

Variavel local: Só existem dentro da instância atual do shell (ou dentro de uma função/script).

3) Para que servem os comandos `env/printenv`? Execute-os e mostre o resultado

Esses comandos exibem as variáveis de ambiente disponíveis no sistema.

[illegible]

4) Dê um exemplo de uma variável de ambiente (shell) e seu significado.

HOME indica o diretório inicial do usuário.

5) Crie uma variável nova no shell atual chamada teste, com o conteúdo “teste”. Mostre em tela o valor da variável criada. Verifique se ela está disponível fora do escopo do shell. Agora faça com que ela seja uma variável global e mostre ela fora do escopo do shell.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050: $ teste="teste"  
2025.1.08.033@suporte-OptiPlex-3050: $ echo $teste  
teste  
2025.1.08.033@suporte-OptiPlex-3050: $ bash  
2025.1.08.033@suporte-OptiPlex-3050: $ echo $teste  
teste  
2025.1.08.033@suporte-OptiPlex-3050: $ export teste="teste"  
2025.1.08.033@suporte-OptiPlex-3050: $ bash  
2025.1.08.033@suporte-OptiPlex-3050: $ echo $teste  
teste  
2025.1.08.033@suporte-OptiPlex-3050:--$
```

- 6) Abra uma nova sessão de shell a partir do mesmo terminal e verifique através do comando ps a existência das duas instâncias de shell rodando ao mesmo tempo. Mostre o valor da variável criada no exercício 5. Em seguida, saia desse shell criado e volte para o anterior. Exclua a variável e mostre que ela foi excluída.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050: $ ps -f | grep bash  
2025.1.+ 16901 16932 0 08:43 pts/0 00:00:00 bash  
2025.1.+ 16941 16901 0 08:43 pts/0 00:00:00 grep --color=auto bash  
2025.1.08.033@suporte-OptiPlex-3050: $ echo $teste  
2025.1.08.033@suporte-OptiPlex-3050:--$
```

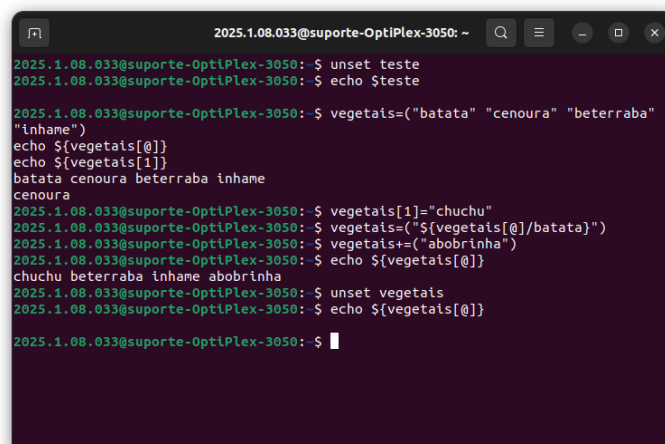
```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050: $ unset teste  
2025.1.08.033@suporte-OptiPlex-3050: $ echo $teste  
2025.1.08.033@suporte-OptiPlex-3050:--$
```

- 7) Arrays de ambiente armazenam múltiplos valores dentro de si. Declare um array de ambiente de nome vegetais e preencha com 4 valores (batata, cenoura, beterraba, inhame). Em seguida, mostre em tela o conteúdo total do array. Depois mostre o conteúdo “cenoura” em tela.

```
2025.1.08.033@suporte-OptiPlex-3050:~$ unset teste
2025.1.08.033@suporte-OptiPlex-3050:~$ echo $teste

2025.1.08.033@suporte-OptiPlex-3050:~$ vegetais=("batata" "cenoura" "beterraba"
"inhame")
echo ${vegetais[@]}
echo ${vegetais[1]}
batata cenoura beterraba inhame
cenoura
```

- 8) No array de ambiente do exercício anterior, altere o valor de cenoura para chuchu. Em seguida, remova batata. Em seguida, adicione abobrinha ao array. Em seguida, exclua todo o array. Em todas as operações, mostre em tela o array para ver se as operações



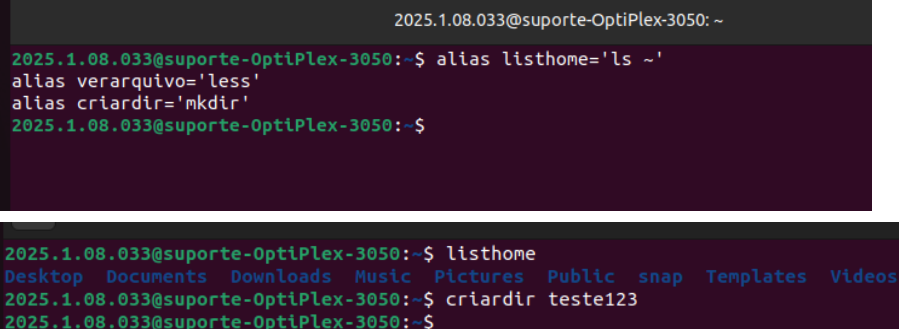
```
2025.1.08.033@suporte-OptiPlex-3050: ~
2025.1.08.033@suporte-OptiPlex-3050:~$ unset teste
2025.1.08.033@suporte-OptiPlex-3050:~$ echo $teste

2025.1.08.033@suporte-OptiPlex-3050:~$ vegetais=("batata" "cenoura" "beterraba"
"inhame")
echo ${vegetais[@]}
echo ${vegetais[1]}
batata cenoura beterraba inhame
cenoura
2025.1.08.033@suporte-OptiPlex-3050:~$ vegetais[1]="chuchu"
2025.1.08.033@suporte-OptiPlex-3050:~$ vegetais=("${vegetais[@]/batata}")
2025.1.08.033@suporte-OptiPlex-3050:~$ vegetais+=("abobrinha")
2025.1.08.033@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}
chuchu beterraba inhame abobrinha
2025.1.08.033@suporte-OptiPlex-3050:~$ unset vegetais
2025.1.08.033@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}

2025.1.08.033@suporte-OptiPlex-3050:~$
```

foram efetivadas.

- 9) Crie aliases para: comando que lista o que contém no diretório do seu home, comando que permite mostrar o conteúdo de um arquivo aos poucos e comando de criação de diretório. Faça testes mostrando que os aliases funcionaram.



```
2025.1.08.033@suporte-OptiPlex-3050: ~
2025.1.08.033@suporte-OptiPlex-3050:~$ alias llisthome='ls ~'
alias verarquivo='less'
alias criardir='mkdir'
2025.1.08.033@suporte-OptiPlex-3050:~$

2025.1.08.033@suporte-OptiPlex-3050:~$ llisthome
Desktop Documents Downloads Music Pictures Public snap Templates Videos
2025.1.08.033@suporte-OptiPlex-3050:~$ criardir teste123
2025.1.08.033@suporte-OptiPlex-3050:~$
```

- 10) Crie uma função chamada `primeirafuncao`, que cria um arquivo dentro do seu home, e escreve no arquivo “Olá, Mundo!”. Faça isso de modo que toda vez que for executado o conteúdo seja anexado e não substituído.

```
2025.1.08.033@suporte-OptiPlex-3050:~$ primeiraFuncao() {
    echo "Olá, Mundo!" >> ~/meuarquivo.txt
}
2025.1.08.033@suporte-OptiPlex-3050:~$
```

- 11) Verifique se a função anterior foi de fato criada e execute-as 5 vezes. Mostre o conteúdo do arquivo após essas execuções, comprovando o funcionamento da função.

```
2025.1.08.033@suporte-OptiPlex-3050:~$ type primeirafuncao
primeirafuncao is a function
primeirafuncao ()
{
    echo "Olá, Mundo!" >> ~/meuarquivo.txt
}
2025.1.08.033@suporte-OptiPlex-3050:~$ primeirafuncao
2025.1.08.033@suporte-OptiPlex-3050:~$ primeirafuncao
2025.1.08.033@suporte-OptiPlex-3050:~$ primeirafuncao
2025.1.08.033@suporte-OptiPlex-3050:~$ primeirafuncao
2025.1.08.033@suporte-OptiPlex-3050:~$ cat ~/meuarquivo.txt
Olá, Mundo!
Olá, Mundo!
Olá, Mundo!
Olá, Mundo!
Olá, Mundo!
2025.1.08.033@suporte-OptiPlex-3050:~$
```