

## Introdução à Ciência da Computação – Lista 4

### Shell script – parte 1

**Nome:** Isabella Cristina da Silveira

**RA:** 2019.2.08.004

**1)** O que é Shell? O que é Shell Scripting? Dê exemplos de shell existentes no padrão Unix.

Shell é a interface de linha de comando que permite ao usuário interagir com o sistema operacional, executando comandos, scripts e programas. Já o Shell Scripting, é a prática de escrever scripts para automatizar tarefas em um sistema Unix. Alguns exemplos de shell existentes no padrão Unix, seriam Bash, Sh, Csh, Tcsh e Zsh.

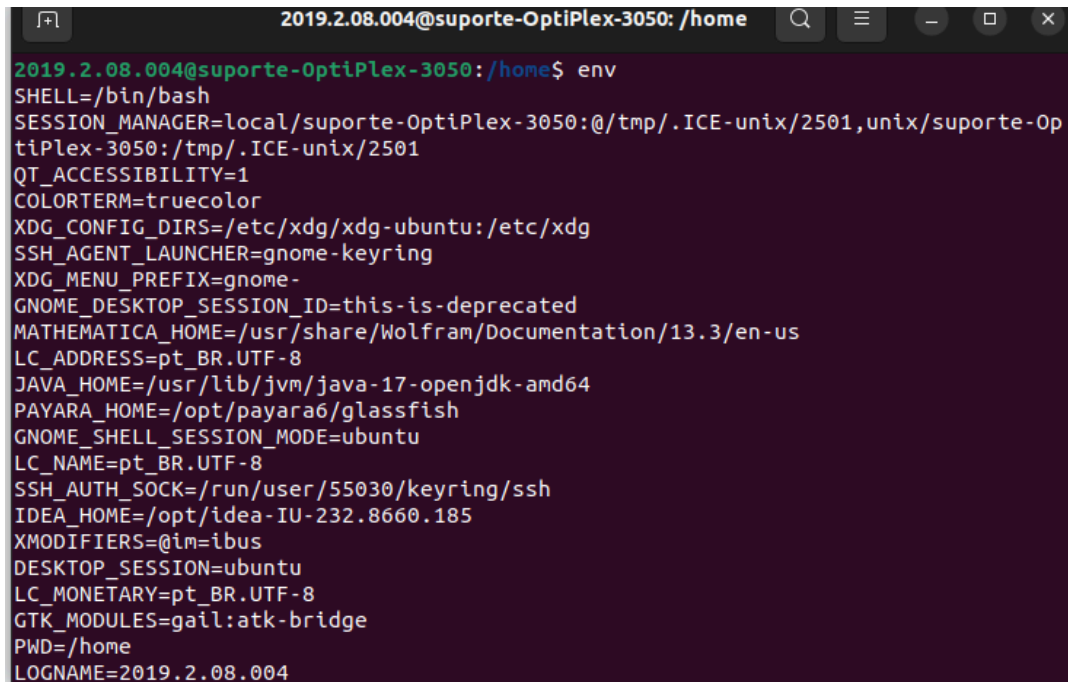
**2)** O que são variáveis de ambiente? Qual a diferença de variáveis globais e locais?

Variáveis de ambiente são variáveis que afetam o comportamento de processos no sistema. Elas armazenam informações como configurações do sistema ou do usuário.

Variáveis globais são variáveis de ambiente que estão disponíveis para todos os processos e shells filhos. Já as variáveis locais são variáveis definidas dentro de um shell ou processo e não estão disponíveis para processos filhos.

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

O comando env serve para exibir ou definir variáveis de ambiente no shell e o printenv serve para exibir todas as variáveis de ambiente ou o valor de uma variável específica.



```
2019.2.08.004@suporte-OptiPlex-3050: /home
2019.2.08.004@suporte-OptiPlex-3050:/home$ env
SHELL=/bin/bash
SESSION_MANAGER=local/suporte-OptiPlex-3050:@/tmp/.ICE-unix/2501,unix/suporte-OptiPlex-3050:/tmp/.ICE-unix/2501
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
MATHEMATICA_HOME=/usr/share/Wolfram/Documentation/13.3/en-us
LC_ADDRESS=pt_BR.UTF-8
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
PAYARA_HOME=/opt/payara6/glassfish
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=pt_BR.UTF-8
SSH_AUTH_SOCK=/run/user/55030/keyring/ssh
IDEA_HOME=/opt/idea-IU-232.8660.185
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=pt_BR.UTF-8
GTK_MODULES=gail:atk-bridge
PWD=/home
LOGNAME=2019.2.08.004
```

```
2019.2.08.004@suporte-OptiPlex-3050: /home
2019.2.08.004@suporte-OptiPlex-3050:/home$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/suporte-OptiPlex-3050:@/tmp/.ICE-unix/2501,unix/suporte-OptiPlex-3050:/tmp/.ICE-unix/2501
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
MATHEMATICA_HOME=/usr/share/Wolfram/Documentation/13.3/en-us
LC_ADDRESS=pt_BR.UTF-8
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
PAYARA_HOME=/opt/payara6/glassfish
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=pt_BR.UTF-8
SSH_AUTH_SOCK=/run/user/55030/keyring/ssh
IDEA_HOME=/opt/idea-IU-232.8660.185
XMODIFIERS=@im=ibus
```

- 4) Dê um exemplo de uma variável de ambiente (shell) e seu significado.  
A variável PATH, que armazena os diretórios onde o sistema busca executáveis.

```
2019.2.08.004@suporte-OptiPlex-3050: ~
2019.2.08.004@suporte-OptiPlex-3050:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin:/usr/lib/jvm/java-17-openjdk-amd64/bin:/opt/apache-maven-3.9.3/bin:/opt/idea-IU-232.8660.185/bin:/opt/payara6/glassfish/bin
2019.2.08.004@suporte-OptiPlex-3050:~$
```

- 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.

```
2019.2.08.004@suporte-OptiPlex-3050: ~
2019.2.08.004@suporte-OptiPlex-3050:~$ teste="teste"
2019.2.08.004@suporte-OptiPlex-3050:~$ echo $teste
teste
2019.2.08.004@suporte-OptiPlex-3050:~$ export teste
2019.2.08.004@suporte-OptiPlex-3050:~$ echo $teste
teste
2019.2.08.004@suporte-OptiPlex-3050:~$
```

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.

```
2019.2.08.004@suporte-OptiPlex-3050: ~  
2019.2.08.004@suporte-OptiPlex-3050:~$ bash  
2019.2.08.004@suporte-OptiPlex-3050:~$ ps  
    PID TTY          TIME CMD  
    6879 pts/0    00:00:00 bash  
    7287 pts/0    00:00:00 bash  
    7294 pts/0    00:00:00 ps  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo $teste  
teste  
2019.2.08.004@suporte-OptiPlex-3050:~$ exit  
exit  
2019.2.08.004@suporte-OptiPlex-3050:~$ unset teste  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo $teste  
2019.2.08.004@suporte-OptiPlex-3050:~$
```

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.

```
2019.2.08.004@suporte-OptiPlex-3050: ~  
2019.2.08.004@suporte-OptiPlex-3050:~$ vegetais=("batata" "cenoura" "beterraba"  
"inhame")  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}  
batata cenoura beterraba inhame  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[1]}  
cenoura  
2019.2.08.004@suporte-OptiPlex-3050:~$
```

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 foram efetivadas.

```
2019.2.08.004@suporte-OptiPlex-3050: ~  
2019.2.08.004@suporte-OptiPlex-3050:~$ vegetais[1]="chuchu"  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}  
batata chuchu beterraba inhame  
2019.2.08.004@suporte-OptiPlex-3050:~$ unset vegetais[0]  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}  
chuchu beterraba inhame  
2019.2.08.004@suporte-OptiPlex-3050:~$ vegetais+=("abobrinha")  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}  
chuchu beterraba inhame abobrinha  
2019.2.08.004@suporte-OptiPlex-3050:~$ unset vegetais  
2019.2.08.004@suporte-OptiPlex-3050:~$ echo ${vegetais[@]}  
2019.2.08.004@suporte-OptiPlex-3050:~$
```

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.

```
2019.2.08.004@suporte-OptiPlex-3050: ~  
2019.2.08.004@suporte-OptiPlex-3050:~$ alias listar_home='ls ~'  
2019.2.08.004@suporte-OptiPlex-3050:~$ alias mostrar_arquivo='less'  
2019.2.08.004@suporte-OptiPlex-3050:~$ alias criar_dir='mkdir'  
2019.2.08.004@suporte-OptiPlex-3050:~$ listar_home  
Desktop Downloads Music Public Templates  
Documents exemplo.txt Pictures snap Videos  
2019.2.08.004@suporte-OptiPlex-3050:~$ criar_dir novo_diretorio  
2019.2.08.004@suporte-OptiPlex-3050:~$ ls  
Desktop Downloads Music Pictures snap Videos  
Documents exemplo.txt novo_diretorio Public Templates  
2019.2.08.004@suporte-OptiPlex-3050:~$
```

```
arquivo exemplo  
exemplo.txt (END)
```

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.

```
2019.2.08.004@suporte-OptiPlex-3050: ~  
2019.2.08.004@suporte-OptiPlex-3050:~$ primeirafuncao() {  
    echo "Olá, Mundo!" >> ~/meu_arquivo.txt  
}  
2019.2.08.004@suporte-OptiPlex-3050:~$ primeirafuncao  
2019.2.08.004@suporte-OptiPlex-3050:~$ cat ~/meu_arquivo.txt  
Olá, Mundo!  
2019.2.08.004@suporte-OptiPlex-3050:~$
```

