



Introdução à Ciência da Computação – Lista 4

Shell script – parte 1

Nome: Filipe Bessa Carvalho

RA: 2024.1.08.009

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

O shell tem como função fornecer uma interface interativa de texto, que pode também interpretar comandos e executar programas. Shell scripting é a programação dentro do ambiente Shell. Alguns shells disponíveis no padrão unix são: bash, csh, e sh.

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

Variáveis de ambiente guardam informações úteis (como informações e preferências do usuário) que podem ser utilizadas para o bom funcionamento do shell. Essas variáveis se dividem em globais e locais. Enquanto variáveis globais podem ser usadas por qualquer subprocesso do shell e outros shells, variáveis locais são específicas e limitadas ao shell atual.

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

Esses comandos mostram as variáveis de ambiente, como no exemplo abaixo:

```

2024.1.08.009@suporte-OptiPlex-3050:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/suporte-OptiPlex-3050:@/tmp/.ICE-unix/2505,unix/suporte-OptiPlex-3050:/tmp/.ICE-unix/2505
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/54305/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/2024.1.08.009
LOGNAME=2024.1.08.009
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=2549
XAUTORITY=/run/user/54305/.mutter-Xwaylandauth.VOPUM2
HOME=/home/2024.1.08.009
USERNAME=2024.1.08.009
IM_CONFIG_PHASE=1
LC_PAPER=pt_BR.UTF-8
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:ml=00:su=37;

```

printenv tem o mesmo resultado.

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

Um exemplo de variável do shell é seconds, que guarda o valor correspondente ao número de segundos passados desde o início do shell.

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.

```

2024.1.08.009@suporte-OptiPlex-3050:~$ teste="teste"
2024.1.08.009@suporte-OptiPlex-3050:~$ echo $teste
teste

```

agora em outro shell

```

2024.1.08.009@suporte-OptiPlex-3050:~$ bash
2024.1.08.009@suporte-OptiPlex-3050:~$ ps
  PID TTY          TIME CMD
 4524 pts/0        00:00:00 bash
 4695 pts/0        00:00:00 bash
 4702 pts/0        00:00:00 ps
2024.1.08.009@suporte-OptiPlex-3050:~$ echo $teste
2024.1.08.009@suporte-OptiPlex-3050:~$

```

tornando teste global

```
2024.1.08.009@suporte-OptiPlex-3050:~$ export teste
2024.1.08.009@suporte-OptiPlex-3050:~$
```

agora em outro shell

```
2024.1.08.009@suporte-OptiPlex-3050:~$ bash
2024.1.08.009@suporte-OptiPlex-3050:~$ ps
  PID TTY          TIME CMD
  4524 pts/0        00:00:00 bash
  4667 pts/0        00:00:00 bash
  4674 pts/0        00:00:00 ps
2024.1.08.009@suporte-OptiPlex-3050:~$ echo $teste
teste
2024.1.08.009@suporte-OptiPlex-3050:~$
```

- 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.

```
2024.1.08.009@suporte-OptiPlex-3050:~$ bash
2024.1.08.009@suporte-OptiPlex-3050:~$ ps
  PID TTY          TIME CMD
  4524 pts/0        00:00:00 bash
  4667 pts/0        00:00:00 bash
  4674 pts/0        00:00:00 ps
2024.1.08.009@suporte-OptiPlex-3050:~$ echo $teste
teste
2024.1.08.009@suporte-OptiPlex-3050:~$
```

```
2024.1.08.009@suporte-OptiPlex-3050:~$ exit
exit
2024.1.08.009@suporte-OptiPlex-3050:~$ unset teste
2024.1.08.009@suporte-OptiPlex-3050:~$ echo $teste

2024.1.08.009@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.

```
2024.1.08.009@suporte-OptiPlex-3050:~$ vegetais=(batata cenoura beterraba inhame)
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[1]}
cenoura
2024.1.08.009@suporte-OptiPlex-3050:~$
```

- 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.

```
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}
batata cenoura beterraba inhame
2024.1.08.009@suporte-OptiPlex-3050:~$ vegetais[1]=chuchu
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}
batata chuchu beterraba inhame
2024.1.08.009@suporte-OptiPlex-3050:~$ unset vegetais[0]
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}
chuchu beterraba inhame
2024.1.08.009@suporte-OptiPlex-3050:~$ vegetais[3]=abobrinha
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}
chuchu beterraba abobrinha
2024.1.08.009@suporte-OptiPlex-3050:~$ vegetais[4]=inhame
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}
chuchu beterraba abobrinha inhame
2024.1.08.009@suporte-OptiPlex-3050:~$ unset vegetais[*]
2024.1.08.009@suporte-OptiPlex-3050:~$ echo ${vegetais[*]}

2024.1.08.009@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.

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ alias lshome='ls ~/'
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ lshome
Desktop Documents Downloads intcc Music NetBeansProjects Pictures Public snap Templates trabalho Videos
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ cd
2024.1.08.009@suporte-OptiPlex-3050:~$ lshome
Desktop Documents Downloads intcc Music NetBeansProjects Pictures Public snap Templates trabalho Videos
2024.1.08.009@suporte-OptiPlex-3050:~$
```

```

2024.1.08.009@suporte-OptiPlex-3050:~$ alias mo=more
2024.1.08.009@suporte-OptiPlex-3050:~$ ls
Desktop Documents Downloads intcc Music NetBeansProjects Pictures Public snap Templates trabalho Videos
2024.1.08.009@suporte-OptiPlex-3050:~$ cd intcc
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ ls
arquivos
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ cd arquivos
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$ ls
agenda.txt arquivo1.txt BD.txt pessoas.txt saidaBD.txt telefones.txt texto.txt xaa xab xac xad xae
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$ mo agenda.txt
Álvaro-4792-9725
João-0234-5542
José-6273-7624
Maria-1134-5673
Úrsula-3572-2615
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$

```

```

2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$ alias md=mkdir
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$ md diretorio
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$ ls
agenda.txt arquivo1.txt BD.txt diretorio pessoas.txt saidaBD.txt telefones.txt texto.txt xaa xab xac xad xae
2024.1.08.009@suporte-OptiPlex-3050:~/intcc/arquivos$

```

- 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.

escolhi dar o nome de `helloworld` à função

```

2024.1.08.009@suporte-OptiPlex-3050:~$ cd intcc
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld(){
> echo 'Hello, World!' >> hw.txt
> }
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ ls
arquivos hw.txt
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ cat hw.txt
Hello, World!
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ cat hw.txt
Hello, World!
Hello, World!
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$

```

- 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.

```
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
[2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ helloworld
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$ cat hw.txt
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!
2024.1.08.009@suporte-OptiPlex-3050:~/intcc$
```