

Capítulo 11 Shell Script - Parte 2

11.1. Objetivos:

Entender como funciona e como criar scripts em Shell

11.2. Introdução à Shell Script



```
$ vi talogado.sh
```

```
# Verifica se um usuário está logado ou não no sistema
```

```
#!/bin/bash
```

```
if who | grep $1
```

```
then
```

```
echo $1 esta logado
```

```
else
```

```
echo $1 nao esta logado
```

```
fi
```

E dê permissão de execução:



```
$ chmod +x talogado.sh
```

Aí é só testar com um usuário:



```
$/talogado.sh leonardo
```

```
leonardo tty1 2006-08-29 11:03
```

```
leonardo esta logado
```

```
$ ./talogado.sh maria
```

```
maria nao esta logado
```

Veja que na execução do script com o usuário leonardo aparece uma linha antes da mensagem de saída, é possível eliminá-la, já que essa não é interessante neste caso.



```
$ vi talogado.sh  
#  
# Verifica se um usuario esta logado ou nao  
#  
#!/bin/bash  
if who | grep $1 > /dev/null  
then  
echo $1 esta logado  
else  
echo $1 nao esta logado  
fi
```



/dev/null é o dispositivo que funciona como uma cesta de lixo no Linux, ou como um buraco negro.

Veja agora como ficou:



```
$ ./talogado.sh leonardo
```



```
leonardo esta logado
```

11.3. Case



Sintaxe do case:

```
case <valor> in
<padrão1>
comando(s)
;; (indica fim de bloco)
<padrão2>
comando(s)
;; (indica fim de bloco)
<padrão3>
comando(s)
;; (indica fim de bloco)
*)
comando(s)
;; (indica fim de bloco)
esac
```

11.3.1. Script exemplo para case:



Alem disso, neste script reforça o uso do if, variável de retorno, quantidade de parâmetros e meta-caracteres.



```
#!/bin/bash
erro=0
if [ $# -ne 1 ]
then
echo "ERRO!!! O certo é $0 caractere"
erro=1
fi
case $1 in
?) ;;
*) echo "ERRO!!! O parametro passado só pode ter um caractere."
erro=2
;;
esac
if [ "$erro" -ne 0 ]
then
exit $erro
fi
case $1 in
[a-z]) echo "Letra minuscula!!!"
;;
[A-Z]) echo "Letra maiuscula!!!"
;;
[0-9]) echo "Numero!!!"
;;
*) echo "Caractere especial!"
;;
esac
```

11.4. Estrutura de repetição o comando while

Apresentando o comando while. Isso vai ser executado fora de um arquivo, mas pode colocar o conteúdo dentro se quiser.



```
#!/bin/bash
n=1
while [ $n -le 12 ]
do
touch arq$n
n=`expr $n + 1`
done
```

Comparando números:



```
num1 -eq num2 : num1 é igual a num2 (equal to)
num1 -ne num2 : num1 não é igual a num2 (not equal to)
num1 -gt num2 : num1 é maior que num2 (greater than)
num1 -ge num2 : num1 é maior ou igual a num2 (greater or equal)
num1 -lt num2 : num1 é menor que num2 (less than)
num1 -le num2 : num1 é menor ou igual a num2 (less or equal)
```

Prática:



```
$ variavel=010
$ test $variavel = 10
```

Para saber se o último comando resultou em verdadeiro ou falso:



```
$echo $?
```

Maneira correta:



```
$ test $variavel -eq 10
```

Reescrevendo o código de forma mais legível:



```
$ vi pedido.sh

#
# Pergunta ao usuário se ele quer listar o diretório corrente
#
#!/bin/bash
resposta=$1
if [ $resposta = S ] || [ $resposta = s ]
then
echo "O conteúdo do diretório corrente é:"
ls
else
if [ $resposta = N ] || [ $resposta = n ]
then
echo "Não quer listar!"
fi
fi
```

Agora, um exemplo muito útil, que é sobre backup. Para que não seja necessário ficar digitando tar o tempo todo, uso shell script para ganhar tempo para fazer outras coisas e ainda evitar a tão temida LER (lesão por esforço repetitivo).



```
# vi backup.sh
```



```
#!/bin/bash
VAR=/mnt/backup-$(date +%Y%m%d).tar.gz
tar cPzf $VAR /etc
```

O valor da variável VAR é /mnt/backup-\$(date +%Y%m%d).tar.gz

Essa linha é onde o backup será guardado. E para que não fique nada desorganizado, o nome do backup juntou-se com o comando date para ele guardar os backups em ordem de data. A última linha é o comando tar já conhecido pedindo para fazer o backup do diretório /etc.

Exemplo: backup-20081023.tar.gz

Então, até aqui na ultima linha o tar com as opções que já vimos antes. O diferente aí é uma opção P.

Pesquise no man:



`$ man tar`



O parâmetro -P serve para usar caminhos absolutos ao invés de relativos, não tirando a / inicial do caminho dos arquivos. Como resultado disso, não será possível restaurar os arquivos do pacote .tar.gz para outro lugar que não seja o mesmo lugar em que eles estavam antes. Se não usarmos -P, podemos restaurar para qualquer outro lugar.