

Miscelânea

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

16 de janeiro de 2025

O comando *eval*

```
$ cano='|'  
$ ls $cano wc -l  
ls: cannot access '|': No such file or directory  
ls: cannot access 'wc': No such file or directory  
$ eval ls $cano wc -l  
5
```

- ▶ Permite avaliar o conteúdo de uma variável como se fosse um comando.
- ▶ Você pode ir construindo um comando em uma variável para depois executá-lo.

O comando *wait*

```
$ sleep 60 &
```

```
[1] 3223
```

```
$ wait 3223
```

- ▶ Bloqueia até o processo termine.
- ▶ Você pode criar vários processos e usar *wait* para ter certeza que todos terminaram.

O comando *trap*

`$ trap "[comando1];[comando2];[comando3]; ... ;[comandoN]" sinais`

0 EXIT	Saída normal
1 SIGHUP	kill -HUP
2 SIGINT	CTRL + C
3 SIGQUIT	CTRL + \
15 SIGTERM	kill PID

- ▶ Permite executar comandos de acordo com os sinais recebidos.
- ▶ Pode ser usado para limpar arquivos temporários.

Funções

```
funcao ()  
{  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN  
    return $var # O valor de var deve ser número  
}
```

Invocação:

```
funcao [parametro1] [parametro2] ... [parametroN]
```

Cada parâmetro dentro da função é tratado por \$1 \$2 \$3 ...

Funções

```
function command_not_found_handle
{
    echo "Erro na linha ${BASH_LINENO[0]}"
    exit
}
```

- ▶ Permite tratar erros de execução de comandos.
- ▶ O comando *exit* encerra a execução do *script*.

O comando *mkfifo*

```
# No terminal 1
$ mkfifo cano
$ ls > cano
# No terminal 2
$ cat < cano
cano sistema.sh trap.sh
```

- ▶ Cria um *pipe* nomeado.
- ▶ Permite comunicação e sincronização entre processos.

O Comando *getopts*

\$ **getopts** [cadeia de opções] variável

- ▶ Se o *script* aceita as opções -a -b -c, então a cadeia de opções deve ser *abc*.
- ▶ Se uma das opções tem argumento, : deve ser colocado depois da letra, a:bc
- ▶ A variável irá armazenar qual opção está sendo tratada, enquanto OPTARG contém o valor passado.

O Comando *getopts*

```
$ cat getopts_exemplo.sh
#!/bin/bash
while getopts "a:bc" OPTVAR
do
    echo $OPTVAR $OPTARG
done
$ ./getopts_exemplo.sh -a 1 -b -c
a 1
b
c
$ ./getopts_exemplo.sh -b -c -a 1
b
c
a 1
```

Execução Passo a Passo

Muitas vezes não é fácil encontrar o erro em *Shell Scripts*.

- ▶ Uma opção é ativar a execução passo a passo.
 - ▶ Coloque `set -x` no início do trecho que deseja analisar.
 - ▶ No fim do trecho desabilite com `set +x`.
- ▶ Se você quiser listar as linhas com mais detalhes.
 - ▶ Ativar com `set -xv`.
 - ▶ Desabilitar com `set +xv`.

Exemplo de Execução Passo a Passo

```
#!/bin/bash
echo "Iniciando execução passo a passo."
set -x
echo "Flag ativado."
read -p "Informe um valor:" entrada
entrada=`expr $entrada + 1`
echo "Valor atualizado: $entrada"
set +x
echo "Flag desativado."
```

Exemplo de Execução Passo a Passo

```
#!/bin/bash
echo "Iniciando execução passo a passo."
set -xv
echo "Flag ativado."
read -p "Informe um valor:" entrada

soma=0
for i in `seq $entrada`
do
    echo "Atualizando entrada..."
    soma=`expr $soma + $i`
done

echo "Valor atualizado: $soma"
set +xv
echo "Flag desativado."
```