

Aula 9 – Operações matemáticas e inicialização de scripts

Na aula passada já vimos um pouco de matemática quando somamos o valor da variável +1 para contarmos o loop:

```
VOLTA=$(( $VOLTA + 1 ))
```

Ou usando o sinal de menos ↓↓↓ (esquece multiplicação e divisão nesta sintaxe)

```
VOLTA=$(( $VOLTA - 1 ))
```

Quando o script precisar literalmente fazer contas igual uma calculadora, usamos a sintaxe abaixo, deixando a conta sempre entre os dois parênteses associando o resultado a uma variável, neste comando não faz diferença escrever com ou sem espaço.

```
#!/bin/bash
CONTA=$((10 * 2))
echo $CONTA
```

Os símbolos matemáticos usados no shell são os que estamos acostumados

+
-
*
/

Acima a variável CONTA é igual ao resultado da operação entre parênteses e no echo eu mostro o resultado, eu poderia por exemplo usar este resultado em algum comando do script.

Uma vez usei este conceito para redundância, se o comando retornasse 64 eu executaria o comando 2, só que apenas uma verificação poderia ter falhas, então coloquei três verificações somei todas elas usando suas variáveis e na lógica de comparação eu dizia que poderia executar o comando **se** o valor fosse igual ou maior que 64, ou seja, se até duas verificações falhassem eu conseguiria o valor 64 o que indicava que pelo menos uma estava funcionando. Abaixo uma aplicação das operações matemáticas usando variáveis.

```
#!/bin/bash
N1="$3"
N2="$2"
CONTA=$(( $N1 * $N2 ))
```

Infelizmente o shell só mostra resultados inteiros se a resposta for fracionária ele mostra como inteiro ou as vezes não funciona.

É difícil dar exemplos práticos, eu usei pouquíssimas vezes contas em script, a não ser que o seu programa seja mais voltado para matemática.

Devemos salientar que se o valor for considerado matemático devemos ter cuidado com as aspas no if, while etc.

Aqui fazemos o comando considerando a condição como um valor matemático	E aqui como um texto, colocando aspas na variável e item a ser comparado
<pre>if [\$CONTA -eq 1];then</pre>	<pre>if ["\$CONTA" = "1"];then</pre>

Podemos ter outras variações que podem dar certo, você sabendo estas 2, vamos complicar pra quê.