Shell Script do zero

Aula 8 – While e Until

Digamos que os dois comandos descritos nesta aula tem um pouco do “if” e um pouco do “for”, um pouco do “for” porque eles também tem como característica rodar em loop e um pouco do “if” porque tem a mesma estrutura e necessitam de uma condição para rodar.

While

While em inglês significa “enquanto”, ou seja, enquanto a condição for verdadeira faça o comando, no “if” se a condição é verdadeira ele roda uma vez, aqui ele vai rodando **enquanto** ela for verdadeira (fica “agarrado” dentro dela até mudar para falsa).

Podemos usar um valor numérico para determinar quantos loops teremos, podemos dar opção do usuário digitar se quer tentar novamente etc. O importante é usar uma variável para “pescar” o while ou until.

Sintaxe

```
while [ Condição ];do
  comando
done
```

Da mesma forma que o if, no while e until usaremos os operadores lógicos de comparação ensinados na aula 3.

Tanto no while quanto no until o valor da variável que ele usará como referência para rodar deve estar definido antes de chegar no comando em questão.

O loop do while pode usar para rodar:

Números

Usando o mesmo conceito do “for” podemos ver que o script abaixo dará cinco voltas: 0, 1, 2, 3 e 4. Eu não ensinei a lidar com matemática, mas basta seguir o conceito abaixo, não coloque aspas na soma e respeite os espaços visíveis (a mesma coisa vale para a condição comparativa).

```
#!/bin/bash

VOLTA=$"0"

while [ $VOLTA -lt 5 ];do

echo $VOLTA

VOLTA=$[ $VOLTA + 1 ]

done
```

Mesmo sendo um valor numérico nós damos este valor usando aspas

Condição: **Enquanto** VOLTA for menor que 5 faça o comando

Comando dentro do loop

Aqui eu somo +1 para contar mais um loop dado: variável $VOLTA é igual ao valor de $VOLTA +1

Fechando o while

Este script diz:

- Volta é igual a zero
- Enquanto $VOLTA for menor que 5 então faça o comando
- Adicione +1 na variável VOLTA

Um exemplo prático foi o que usei no proteccontinuo, eu perguntava ao usuário quantos players ele desejava adicionar no programa e quando o script fosse criar os players ele pegava a variável que usei com o usuário, então se o usuário respondeu 3 o while dava 3 loops, repetindo o comando dentro do while 3 vezes e consequentemente criando o proteccontinuo para 3 players.

Texto

Podermos pegar textos para dar voltas no while usando a saída de um comando por exemplo, eu faço um programa de verificação e **enquanto** a saída deste comando for determinada palavra o while vai rodando até que ela mude e saia do loop. As variações são infinitas, mais para frente aprenderemos a filtrar textos/saídas de uma forma que possamos resumi-los a uma linha, uma palavra etc.

A seguir usaremos um loop que dará voltas e voltas até que o usuário decida passar adiante.

```
#!/bin/bash

VOLTA=$"sim"

while [ "$VOLTA" = "sim" ];do

echo "Digite sim se deseja tentar novamente ou con para continuar"
read VOLTA

done
```

Garantindo que ele entrará no loop, poderiamos perguntar isto ao usuário também

O script poderá fazer algo que só o usuário pode decidir quando prosseguir, então ele responde continuar e o while automaticamente sairá do loop.

É importantíssimo entender que o “movimento” para tirar o while ou until do loop deve estar dentro dele, senão rodará “para sempre”.

Until

Until significa até → até que determinada condição seja verdeira execute, ou seja, ele executa se a condição for falsa e só termina quando ela for verdadeira. Então eu posso usar uma variável de valor 5 e **até** que ela se torne 10 vai rodando e dentro do until usaremos a mesma sintaxe de soma que usamos no while. O dado que determina o loop também pode ser texto, mas não acho interessante, o importante é você usá-lo em qualquer situação em que ache necessário usar o “**até”**.

Sintaxe

```
until [ Condição ];do
  comando
done
```

Podemos ter mais de uma condição tanto no while quanto no until usando os conectores “E” e “OU” apresentados na aula 5.

Ele é idêntico ao while, tendo apenas a sua **lógica invertida,** tudo que abordamos com o while anteriormente vale para o until, então não explicarei novamente, vou dar um exemplo:

```
#!/bin/bash

VOLTA=$"0"

until [ $VOLTA -gt 5 ];do

echo $VOLTA

VOLTA=$[ $VOLTA + 1 ]

done
```

Até que $VOLTA seja maior que 5 então faça o comando. Ele vai rodando até que a condição se torne verdadeira, ou seja, **até** $VOLTA ser maior que 5.

As suas voltas serão: 0, 1, 2, 3, 4 e 5

Exercício 5 – Pedindo senha ao usuário

O que o script faz:

- Dá boas vindas
- Pede senha ao usuário

A senha pode ser 123

- A senha sendo correta ele executa o comando → echo “senha correta” e sai
- Caso contrário o usuário tenta novamente sem sair do script (loop)
- São até 5 tentativas e depois das 5 o script mostra “tentativas esgotadas” e sai

O desafio aqui é a lógica e saber quais comandos colocar dentro do loop.

Este script terá utilidade quando aprendermos sobre funções, já que ela é um atalho que nos joga para determinada parte do script, então supondo que façamos um programa de código fechado, o usuário só conseguiria executar determinada parte do script se digitasse a senha corretamente e o atalho função estaria dentro do while ou until.

Exercício pronto → http://www.mediafire.com/download/xv6k3d9653xvubj/senha

Até a próxima, onde aprenderemos sobre operações matemáticas e inicialização de scripts