

Shell Script do zero

Aula 10 – Comando Case

É muito mais fácil criarmos menus com um comando próprio, na programação nós temos o “case” que facilita em muito a construção destas estruturas, com ele podemos colher os dados de uma variável que indicará qual opção o case deve executar.

Estrutura do case

Supondo que criamos um menu com 3 opções para o usuário escolher, o case executa o comando de acordo com o valor recebido pela variável.

case \$VARIÁVEL in **Caso** o valor da variável esteja **em** → 1, 2 ou 3 faça os comandos correspondente a seu valor

1)
Comando;; O número ou palavra da opção fica antes do parênteses e do lado ou abaixo ficam os comandos a serem executados (1 em cada linha).

2)
Comando;;
3)
Comando;; Depois de adicionar todas as linhas daquela opção devemos colocar no final do ultimo comando ponto e virgula duas vezes para fechá-la.

esac Aqui fechamos o case escrevendo ele ao contrario

Exemplo na prática

```
echo "Qual é o comando do seu Screensaver?"  
echo  
echo "1- mate-screensaver"  
echo "2- gnome-screensaver"  
echo "3- XScreensaver"  
echo "4- Outro"
```

Mostrando o menu para o usuário

```
read distro  
echo  
case $distro in
```

Colhendo o valor da variável e abrindo o case

```
1)  
SCREN="$mate-screensaver";;
```

```
2)  
SCREN="$gnome-screensaver";;
```

```
3)  
SCREN="$xscreensaver";;
```

```
4)  
echo "Digite o comando"  
read SCREN;;
```

Repare que na opção 4 temos dois comandos, o ponto e virgula só consta no ultimo comando.

```
*)  
echo "Opção invalida"  
exit;;
```

Aqui eu coloquei a opção “*”, ou seja, se o usuário digitar algo que não consta nas opções ele cai aqui

```
esac
```

Não existem limites na quantidade de linhas que colocaremos nas opções, eu já cheguei a colocar verdadeiros scripts dentro de cada uma.

É isto, não temos muito o que falar do case, com ele basta criarmos condições para que a variável receba algum valor antes de executá-lo e sempre repetir sua sintaxe:

- Abertura do comando → `case variável_usada in`
- Opções que ficam antes dos parênteses → `número_ou_texto)`
- Terminando os comandos da opção, colocar ponto e vírgula 2 vezes no final do último comando
- Fechá-lo escrevendo case ao contrário → `esac`

Podemos precisar mostrar na tela textos usando o echo que contenham acentos ou símbolos, neste caso é obrigatório colocá-los entre aspas, senão é erro atrás de erro, eu sempre coloco entre aspas assim é menos uma coisa para se preocupar (as variáveis que ficarão dentro de aspas fazem a sua função de mostrar seu conteúdo normalmente)

Exercício 6 – Pequeno desafio de lógica

A variável recebe um número digitado pelo usuário

São 3 rodadas no total e em cada uma delas o script multiplica o número dado pelo usuário com o número indicado abaixo:

1º rodada → 10

2º rodada → 20

3º rodada → 30

Então se o usuário digitou 3, serão 3x10 3x20 e 3x30 mostrando na tela um resultado a cada loop, o desafio aqui é fazer uma multiplicação diferente a cada rodada já que não podemos prever o número que será digitado pelo usuário.

Exemplo de como deve aparecer na tela:

```
luiz@linuxmint ~ $ ./6
Por favor digite o número a ser processado
30
Rodada 1 - 30 multiplicado por 10 é igual a 300
Rodada 2 - 30 multiplicado por 20 é igual a 600
Rodada 3 - 30 multiplicado por 30 é igual a 900
```

Existem várias formas de se fazer este pequeno script, aqui usaremos o **while** e **case** para alcançarmos os nossos objetivos. Se você tiver dificuldades saiba que usaremos 90 % do tempo para pensar e 10 % para escrever o script.

Exercício pronto → <http://www.mediafire.com/download/xz2zwbk58x67mee/6>

Até a próxima !!!