

Shell Script

Básico do básico

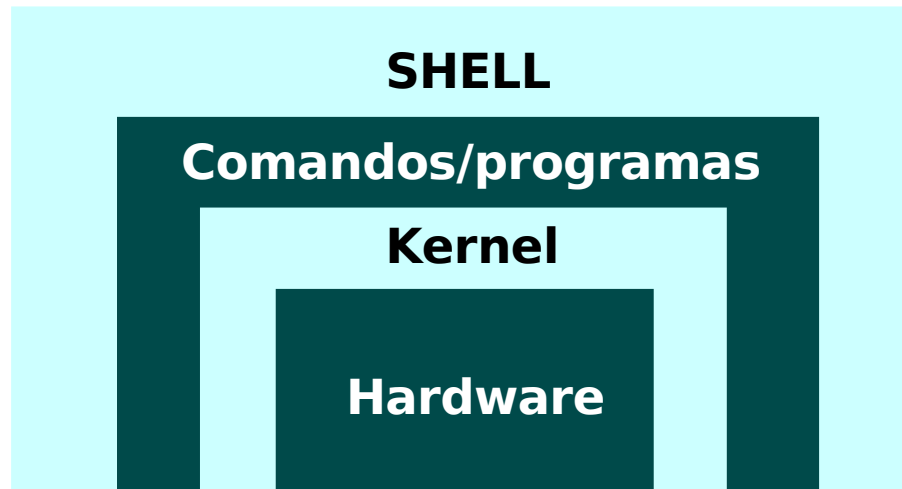


Shell Script – Básico do básico

Os sistemas Unix-like, como o GNU/Linux, possuem camadas.

Estas camadas são o hardware, o kernel, os programas/comandos e o shell.

O shell é a camada mais externa do sistema e tem ligação íntima (comunicação) com os comandos e programas, os comandos e programas têm ligação íntima com o kernel e este com o hardware.



Shell Script – Básico do básico

O shell é o responsável pela interação entre o usuário e o sistema operacional, pois ele é que interpreta os comandos e os traduz para uma linguagem simples e inteligível para o kernel.

Devido a essa função, o kernel tem seu desempenho melhorado, pois é menos um trabalho para o mesmo executar.

O shell se parece visualmente com o MS-DOS da Microsoft. Já sabemos o que é Shell, agora definirei o que é Shell Script.

Script é um arquivo de texto que possui uma sequência de instruções e comandos, estes são executados linha a linha. Podemos fazer uma comparação grotesca do Shell Script com os arquivos .bat do DOS.



Shell Script – Básico do básico

A vantagem de se programar em Shell Script é automatizar tarefas rotineiras, como backup, instalação ou remoção de programas.

Por exemplo, se eu tenho 100 usuários para adicionar no sistema posso fazer um script em shell para realizar essa tarefa em segundos!

Bourne Again Shell: é o shell desenvolvido para o projeto GNU usado pelo GNU/Linux, é muito usado pois o sistema que o porta (GNU/Linux) evolui e é adotado rapidamente. Possui uma boa portabilidade, pois possui características do Korn Shell e C Shell. É representado por “**bash**”. O nosso estudo será focado neste.



Shell Script – Básico do básico

Funções do Shell

Analisar dados a partir do prompt (dados de entrada);

Interpretar comandos;

Controlar ambiente Unix-like (console);

Fazer redirecionamento de entrada e saída;

Execução de programas;

Linguagem de programação interpretada.

Conhecimentos básicos

O uso da tralha (#)

A tralha representa em algumas linguagens de programação um comentário, com o Shell Script não é diferente.



Shell Script – Básico do básico

Um script em Shell é iniciado com a seguinte linha:

```
#!/bin/SHELL_EM_USO
```

Como saber qual shell estou usando?

```
$ echo $SHELL
```

Para o GNU/Linux:

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

Esta linha acima indica o caminho (path) para o interpretador que será usado no script.



Shell Script – Básico do básico

Crase

As crases são usadas para dar prioridade a um comando, veja um exemplo:

```
$ echo "A versão do kernel do `uname -o` é `uname -r`"
```

Saída:

A versão do kernel deste GNU/Linux é 2.6.15

Se você tirar as crases, veja a saída:

A versão do kernel deste uname -o é uname -r



Shell Script – Básico do básico

Variável

Uma variável é representada por \$ (cifrão).

Exemplo de variável:

```
# guarda_roupa=camiseta  
# echo $guarda_roupa
```

Saída:
camiseta



Shell Script – Básico do básico

Permissão para execução de script

Depois de editar um novo script, é necessário que modifiquemos a permissão deste arquivo, senão este não poderá ser executado, veja o porquê:

```
$ ls -l
```

```
-rw-r--r-- 1 leo users 0 2006-05-20 13:20 codigo_fonte
```

```
$ chmod +x codigo_fonte
```

```
$ ls -l
```

```
-rwxr-xr-x 1 leo users 0 2006-05-20 13:20 codigo_fonte*
```

Para executar:

```
$ ./codigo_fonte
```



Shell Script – Básico do básico

Ponto e vírgula (;)

O ponto e vírgula (;) serve para separar comandos.

Exemplo:

```
# mkdir guardaroupa;cd guardaroupa;touch camiseta;touch bermuda
```

Este conjunto de comandos diz:

Crie um diretório chamado "guardaroupa", entre nele e crie os arquivos "camiseta" e "bermuda" vazios.



Shell Script – Básico do básico

Matemática no Shell

Operadores aritméticos:

+ Soma

- Subtração

* Produto

/ Divisão

% Resto da divisão

O comando **expr**

Para fazer um cálculo é necessário usar o comando **expr**.

Exemplo:

\$ **expr** 20 + 5

Atenção: Após cada cadeia de caracteres no uso do **expr** deve ser colocado um espaço em branco (tab).



Shell Script – Básico do básico

Matemática no Shell

O comando let

Realiza aritmética com números inteiros.

Sintaxe:

```
$ let <expressão>
```

Exemplo:

```
$ a=10
$ b=10
$ let c=a+b
$ echo $c
20

$ let a++
$ echo $a
11

$ let b--
$ echo $b
9
```



Shell Script – Básico do básico

O comando `expr + length`

Para pegar o número de caracteres de uma string, use o comando **length**.

Veja:

```
$ expr length "4Linux & Hackerteen"
```

Saída: 19

Observação: o shell só conta o que está dentro das aspas, inclusive os espaços em branco.

Usando a calculadora `bc` do GNU/Linux e o comando `scale`.

O comando `scale` serve para determinar o número de casas decimais.

```
$ echo "scale=1; (25 / 3) + 20" | bc
```

Resultado: 28.3



Shell Script – Básico do básico

Trabalhando com parâmetros

Você já pensou como se passa parâmetros para um programa em shell script???

Um parâmetro é representado por **\$n**, onde **n** é a posição do caractere ou a posição de um conjunto de caracteres.

\$1 representa o primeiro caractere ou o primeiro conjunto de caracteres;

\$/meu_script L

\$/meu_script Linux

\$2 representa o segundo caractere ou o segundo conjunto de caracteres e assim por diante.

\$/meu_script L I

\$/meu_script L INUX

Cada parâmetro é separado por espaço.



Shell Script – Básico do básico

Na prática, veja como é fácil:

Vamos supor que eu tenho um programa chamado monte_nome e eu quero exibir o nome montado após receber letra por letra. **Veja o script:**

```
#####  
# Script para montar nomes  
# Este script recebe seis parâmetros  
#####  
#!/bin/bash  
echo $1$2$3$4$5$6  
# Fim do script
```

```
$ ./monte_nome 4 L I N U X
```

(Note que entre cada parâmetro há um espaço)

Saída: 4LINUX



Shell Script – Básico do básico

Mas pense neste programa anterior que fiz, é péssimo pois está limitado a seis caracteres apenas.

É simples arrumar isso, veja:

```
# Script para montar nomes  
# Este script recebe "n" parâmetros  
#!/bin/bash  
echo $*  
# Fim do script
```

```
$ ./monta_nome H A C K E R T E E N
```

Saída: HACKERTEEN

O **\$*** recebe todos os parâmetros passados.

O parâmetro **\$0** representa o nome do próprio programa.

O parâmetro **\$#** indica a quantidade de parâmetros passados ao script.



Shell Script – Básico do básico

Como ler dados em Shell Script

Usamos o comando **read** para ler dados de arquivos ou diretamente do teclado.

Sintaxe:

```
read var1 var2 var3 ... varN
```

No terminal faça:

```
$ read Linux
```

```
Debian
```

```
$ echo $Linux
```

```
Debian
```



Shell Script – Básico do básico

Bibliografia

Programação Shell Linux 3ª edição

Autor: Julio Cezar Neves

Editora: BRASPORT

Linux Guia do Administrador do Sistema

Autor: Rubem E. Ferreira

Editora: Novatec

