

Notas de Aula gentilmente disponibilizadas pelo aluno Pedro Henrique

Resumo Importante sobre o Curso de LPI-1

Aulas 8 – Tópico 103.1 - Trabalhando na Linha de Comando – Shell, Bash, Echo, Type, Path

O que estudamos nesta aula?

Tipos de SHELL	2
O comando echo	2
O comando type	3
O comando PATH.....	3
 Caminho absoluto:	4
 Caminho Relativo ou Parcial:	4
 Dentro do diretório:	5

Tipos de SHELL

O curso LPI-1 cobra o **Shell Bash**. Os principais tipos de Shell são:

- **Bourne Shell:** é o shell padrão para Unix, ou seja, a matriz dos outros shells. É representado por "sh". Foi desenvolvido por Stephen Bourne, por isso Bourne Shell.
- **Korn Shell:** este shell é o Bourne Shell evoluído, portando todos os comandos que funcionavam no Bourne Shell funcionarão neste com a vantagem de ter mais opções. É representado por "ksh".
- **C Shell:** é o shell mais utilizado em BSD, e possui uma sintaxe muito parecida com a linguagem C. Este tipo de shell já se distancia mais do Bourne Shell, portanto quem programa para ele terá problemas quanto a portabilidade em outros tipos. É representado por "csh".
- **Bourne Again Shell:** é o shell desenvolvido para o projeto GNU usado pelo GNU/Linux, é muito usado pois o sistema que o porta 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 estará focado neste.**

Fonte: <https://www.vivaolinux.com.br/artigo/Introducao-ao-uso-de-Shell-Script?pagina=2>

Para verificar qual SHELL estamos usando, basta dar o comando abaixo, ele irá imprimir na tela o caminho do shell bash “/bin/shell”:

O comando echo

```
$ echo $SHELL
```

echo – O comando serve para imprimir informações na tela. Em conjunto com o símbolo de redirecionamento de saída >>, estudado mais à frente, também é utilizado para concatenar informações para dentro de um arquivo. Exemplo:

```
$ echo "Este é o curso de lpi-1" >> /home/lpi1/Documents/teste.txt
```

No exemplo acima, introduzimos a frase no arquivo teste.txt, mesmo o arquivo não existindo no diretório, ele é criado automaticamente ao executarmos o comando.

O comando type

No Linux existem comandos que são internos do shell que já são embutidos no programa SHELL e também comandos que são externos ou migrados de outras aplicações que estão em nosso sistema. Para sabermos se um comando é de origem SHELL para usarmos o comando **type**, exemplo:

```
$ type echo
```

Após esse comando ele nos retornará uma mensagem nos dizendo se é ou não um comando interno do shell, ou seja, já está em sua compilação de origem.

```
lpil@linux:~/Desktop$ type echo
echo is a shell builtin
lpil@linux:~/Desktop$
```

Quando fazemos o type apontando para outro comando que é externo, ele nos mostra a localização do comando:

```
lpil@linux:~/Desktop$ type tar
tar is /bin/tar
lpil@linux:~/Desktop$
```

Um comando mesmo que externo após digitado muitas vezes o Linux nos cria um cache interno como é o caso do comando “clear”. O comando “clear” é um comando externo.

Após algumas vezes digitado ele passa a ser hashed. cacheado pelo sistema Linux:

```
lpil@linux:~/Desktop$ type clear
clear is hashed (/usr/bin/clear)
lpil@linux:~/Desktop$
```

O comando PATH

O PATH serve para indicar para o sistema o caminho dos comandos externos dentro do Linux. Quando damos o comando abaixo ele nos mostra os diretórios em que os programas e comandos estarão localizados:

```
$ echo $PATH
```

Quando um comando é acionado ele procura em cada diretório onde está o programa correspondente. Os diretórios estão separados por “:”.

```
lpil@linux:~/Desktop$ echo $PATH
/home/lpil/bin:/home/lpil/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
lpil@linux:~/Desktop$
```

Há também formas de chamar um comando quando o ele não é interno e nem está presente dentro dos diretórios acima dos programas externos. Para fazer isso podemos fazer das seguintes formas:

Usaremos o arquivo do exercício disponibilizado no curso como é mostrado na aula:

Caminho absoluto:

Dessa forma ele executará normalmente, pois indicamos o path exato de onde está o programa que queremos executar.

```
$ /home/lpil/Exercicios/Script_Exemplo.sh
```

```
lpil@linux:~$ /home/lpil/Exercicios/Script_Exemplo.sh
Este é um Script de Teste

Qua Jun  7 02:21:35 BRT 2017

Fim do Script
lpil@linux:~$
```

Caminho Relativo ou Parcial:

Neste caso, se você já estiver em uma pasta que está no mesmo caminho que o script que deseja executar basta continuar o comando até o caminho do script. Perceba que, estamos dentro da pasta do usuário **/home/lip1**, podemos conferir com o comando “**pwd**”, o script se encontra dentro de **Exercicios/Script_Exemplo.sh** que já é parte do caminho do script:

```
lpil@linux:~$ pwd
/home/lpil
lpil@linux:~$ Exercicios/Script_Exemplo.sh
Este é um Script de Teste

Qua Jun  7 02:23:17 BRT 2017

Fim do Script
lpil@linux:~$
```

Dentro do diretório:

Neste caso, usaremos “./” **que quer dizer: neste diretório**. Para executar o comando dessa forma temos que estar dentro do diretório em que o script se encontra, e lá, fazemos da seguinte forma para termos sucesso no comando:

\$./Script_Exemplo.sh

```
lpil@linux:~/Exercicios$ pwd
/home/lpil/Exercicios
lpil@linux:~/Exercicios$ ./Script_Exemplo.sh
Este é um Script de Teste

Qua Jun  7 02:26:15 BRT 2017

Fim do Script
lpil@linux:~/Exercicios$
```