

Aula 5

5.1 O ambiente de operação

Em todo sistema Linux ou Unix, a principal ferramenta de operação é o interpretador de comandos. Por sua vez, este nada mais é do que um programa que faz a interface dos comandos do usuário com o sistema. Existem vários programas conhecidos como *shells* ou interpretadores de comandos, dentre eles, destacam-se o *csch*, *tch*, *zsh*, *bash*, etc.

O mais utilizado, dentre os interpretadores citados, é o *bash* e atualmente é tido como base para as instruções da maior parte da literatura sobre sistemas Linux. O *bash* possui muitas funções e atalhos interessantes:

- TAB – completa automaticamente a palavra/comando ou exibe possibilidades existentes.
- CTRL+d – encerra sessão.
- CTRL+e – vai para o final da linha.
- CTRL+a – volta para o início da linha.
- CTRL+u – limpa o texto digitado a esquerda do *prompt*.
- UP (seta para cima) – navega para o comando anterior no histórico de comandos.
- DOWN (seta para baixo) – navega para o próximo comando no histórico de comandos.
- CTRL+r – procura um comando no histórico de comandos.
- CTRL+l – limpa a tela.

Toda vez que uma nova sessão é aberta através de interface de linha de comando; o *shell* é invocado e fica executando à espera de comandos. Em cada uma dessas sessões, diz-se que existe um ambiente ou contexto para os quais sempre vão existir variáveis de controle e de configurações. As variáveis que determinam as características e atributos de um ambiente são chamadas de variáveis de ambiente.

Em cada sessão ativa é possível visualizar as variáveis de ambiente através do comando *printenv*. Conhecendo cada uma dessas variáveis, e também seu significado, é possível alterar seus atributos e conseqüentemente o comportamento de maneira conveniente.

As variáveis de ambiente são escritas em caracteres maiúsculos e precedidas pelo símbolo \$. Normalmente, os nomes reservados dessas variáveis tem palavras vinculadas ao seu significado e são facilmente distinguidas.

A classificação das variáveis se dá quanto ao escopo delas. Podendo ser globais ou locais. As locais são aquelas válidas somente no escopo corrente, em geral uma função local ou uma sessão. Enquanto que as globais, como o nome já diz, são válidas para quaisquer subprocessos e sessões herdeiras ativas.

Alguns comandos importantes para a manipulação de variáveis de ambiente são:

- *export* : atribui valor a uma variável de ambiente
Ex: export EDITOR=/bin/vi

- *unset* : exclui o valor de uma variável de ambiente.
Ex: unset EDITOR

- *echo* : exibe o eco na saída padrão para o parâmetro informado. Muito usado também para exibir o valor de uma variável na tela.
Ex : echo \$EDITOR

Algumas variáveis importantes são:

- PATH: indica o local onde o interpretador de programas deverá procurar pelos executáveis e comandos fornecidos.
- SHELL: indica o interpretador utilizado. O mais comum é o *bash*.
- HOME: indica o diretório pessoal do usuário que usa o ambiente.
- TERM: indica o tipo de terminal utilizado. O mais utilizado é *vt100*.
- PS1 e PS2: indica os caracteres de *prompt* exibidos na tela.

5.2 Scripts

Todos interpretadores executam interagindo com o sistema e devolvendo respostas para a saída padrão. Por isso, é possível utilizarmos esses interpretadores para executarem comandos em lotes de forma automatizada.

Esses lotes, poderíamos dizer que, são arquivos contendo linhas de comandos passíveis de execução pelo interpretador. Com isso, basta adicionar a um arquivo vazio, as linhas dos comandos que se deseja executar, e após isso, acionar esse arquivo de texto final que contém essas linhas de instruções através de um *shell*.

Um exemplo seria:

```
#!/bin/bash

ls -la /etc
grep user /etc/passwd
cp -r /etc/* /tmp/
echo "Saindo do script"
```

O script acima (de nenhuma utilidade prática, é bem verdade) iria executar a listagem do diretório *etc*, em seguida filtraria a palavra *user* do arquivo *passwd* e por fim copiaria todo o conteúdo de *etc* recursivamente para o diretório *tmp*.

A última linha do script iria imprimir na tela a mensagem "*Saindo do script*" e retornaria a execução ao *prompt* do shell.

Para executar esse arquivo de *script* é necessário que o mesmo seja um arquivo executável. Para tal, costuma-se utilizar a alteração de permissão através do comando: *chmod u+x*. Outra definição característica comum fica na primeira linha do *script*, e em geral, contém um comentário indicando qual o interpretador escolhido. Essa linha iniciada pelo símbolo #, na verdade indica que o arquivo é um *script* e deve ser interpretado pelo *shell* descrito nessa linha. Portanto, todos os *scripts*, costumam possuir essa linha e também costumam ter permissão para execução.

Com *scripts*, é possível criarmos programas ou rotinas para o sistema de maneira simples e rápida. Basta saber o que o *bash* e outros interpretadores aceitam em forma de comandos de decisão e instruções lógicas como *if*, *while*, *for*, *case*, *select*, entre outros.

Com essas instruções e todos os comandos dos interpretadores é possível criar *scripts* poderosos e com isso fazer uso dos mesmos na operação e administração do sistema.

Alguns exemplos mais práticos de *scripts* podem ser vistos a seguir.

```
#!/bin/bash
echo -n "Digite seu nome!"
read nome sobrenome
echo "Prazer em conhecê-lo $nome !"
### Fim do script
```

```
#!/bin/bash
fatorial()
{
    echo -n "Forneça um numero"
    read n
    let A=1
    while (($n > 0))
    do
        let A=A*n
        let n=n-1
    done
    echo $A
}
```

```
fatorial
### Fim do script
```

O primeiro *script* lê uma cadeia de caracteres onde o usuário deve digitar seu nome e a rotina devolve para a tela uma saudação com o valor do nome armazenado na variável *\$nome*.

O segundo, já mais complexo, chama a função fatorial que lê o valor de um número inteiro *n* qualquer, processo o fatorial, e imprime o valor do resultado do fatorial de *n* na tela.

Uma boa literatura e referência para a programa em *scripts* no Linux, pode ser encontrada no livro de Júlio Cezar Neves: Programando em Shell ou através do Guia completo online (<http://www.museum.state.il.us/ismdepts/library/linuxguides/abs-guide/>).

5.3 Exercícios

1 – Crie um *script* que solicita ao usuário um comando/aplicativo instalado e em seguida solicita também um alias para esse comando. Ao final, o *script* deve ativar o funcionamento do alias informado pelo usuário.

2 – Crie um *script* que lista o conteúdo de um diretório fornecido pelo usuário. Baseado nesse diretório, o *script* deve copiar todos arquivos com extensão *.txt* e armazená-los no diretório */tmp/backup/* com os mesmos nomes, entretanto adicionando as extensões *.bkp*.

3 – Altere a variável de sinal de *prompt* do seu sistema de forma que a mesma receba o seu nome completo (nome do aluno). Informe os comandos para isso.

5.4 Bibliografia

Este documento é resenha retirada do capítulo 6 de: Ferreira, E. Rubem. Guia do Administrador Linux. 1 edição, 2003, Novatec Editora, São Paulo.