

Comandos Básicos Linux e Introdução ao Shell Script

Professores: Mário Lemes e Uyara Ferreira

Instituto Federal de Educação, Ciência e Tecnologia Goiás - Câmpus Formosa

VI Semana de Educação, Ciência e Tecnologia (SECITEC)

18 de Novembro de 2015

Sistemas Operacionais

O que é um Sistema Operacional?

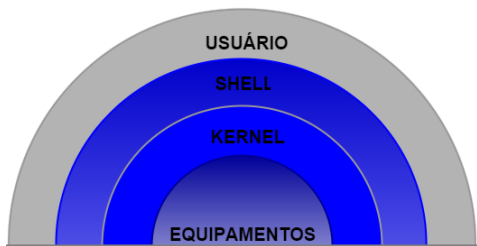
- É uma camada de programas, que faz a interface entre o *hardware* e os programas que executam tarefas para os usuários.

Introdução ao Linux

Comandos Básicos de Linux: Parte I
Comandos Básicos de Linux: Parte II
Introdução ao Shell Script
Comandos Básicos de Shell Script
Considerações Finais
Agradecimentos
Contato com os autores

O que é Sistema Operacional?

História do Linux
Distribuições Linux
Interface Gráfica × Linha de Comando
Árvore de Diretórios



História do Linux

- O que é Software Livre?
- A licença GPL
- O que é Linux? Ou seria GNU/Linux?

Distribuições Linux

- As distribuições são sistemas GNU/Linux que estão previamente agrupados e configurados, prontos para serem instalados. Podem surgir em diversos tipos, disponíveis gratuitamente até totalmente comerciais.
- Todas distribuições oferecem diversos tipos de aplicativos, utilitários e ferramentas de gerenciamento para facilitar o uso do Linux.

Interface Gráfica × Linha de Comando

- No início da informática, só existiam as linhas de comandos, os comandos eram simples e diretos, mas era necessário decorá-los, logo depois veio a interface gráfica aliada ao *mouse*.
- Apesar da interface gráfica, as linhas de comandos são muito importantes para instalações mais complexas e é ideal para servidores, no qual a interface gráfica pode fazer uso de recursos de *hardware* desnecessariamente ou até mesmo incorporar alguma vulnerabilidade ao sistema.

Árvore de Diretórios

- Os diretórios armazenam e organizam arquivos e outros diretórios.
- As estruturas de arquivos e diretórios podem ser vistas como uma árvore, onde o início da ramificação é a raiz (root) ou barra (/), de onde partem todos os outros diretórios.

Estrutura Básica de um Comando

- Usualmente um comando possui a seguinte estrutura básica:

Acessar Diretórios (cd)

Utilizado para acessar os diretórios do sistema.

Exemplo:

- `cd /home/aluno`

Listar Arquivos e Diretórios (ls)

Utilizado para listar arquivos e diretórios.

Exemplo:

- `ls /home/aluno`

Descobririndo o Diretório Corrente (pwd)

Utilizado para indicar em que lugar do sistema o usuário se encontra, ou seja, o diretório corrente.

Exemplo:

- pwd

Criar Diretórios (mkdir)

Utilizado para criar novos diretórios.

Exemplos:

- `mkdir /home/aluno/minicursolinux`
- `mkdir /home/aluno/minicursoshellscript`

Criar Arquivos Vazios (touch)

Utilizado para criar arquivos vazios.

Exemplos:

- touch /home/aluno/minicursolinux/teste0.txt
- touch /home/aluno/minicursolinux/teste1.txt
- touch /home/aluno/minicursoshellscript/teste2.txt
- touch /home/aluno/minicursoshellscript/teste3.txt

Cópia de Arquivos e Diretórios (cp)

Utilizado para copiar arquivos ou diretórios.

Exemplos:

- `cp /home/aluno/minicursolinux/teste0.txt /home/aluno/Documentos`
- `cp -r /home/aluno/minicursolinux /home/aluno/`

Mover e Renomear Arquivos e Diretórios (mv)

Utilizado para mover arquivos e diretórios caso o caminho, origem/destino apontados sejam diferentes, ou para renomeá-los, caso o caminho, origem/destino, apontados sejam o mesmo.

Exemplo:

- `mv /home/aluno/Documentos/teste0.txt /home/aluno`

Remover Arquivos e Diretórios (rm)

Utilizado para remover arquivos e diretórios, sendo este vazio ou com subdiretórios.

Exemplos:

- `rm -i /home/aluno/teste0.txt`
- `rm -f /home/aluno/teste1.txt`

Limpar a Tela (clear)

Utilizado para limpar a tela de comandos.

Sintaxe:

- clear

Exibir Histórico de Comandos (history)

Utilizado para exibir os comandos digitados pelo usuário.

Sintaxe:

- history

Editores de Texto

Pela linha de comando é possível utilizar editores de texto, assim como o Notepad do *Windows*, no Linux existem vários editores, alguns deles são:

- pico
- nano
- vim
- vi

Pesquisando Arquivos com find

Utilizado para encontrar no sistema algum arquivo ou diretório.

- `find / -iname teste0.txt`

Exibir e Alterar a Data/Hora do Sistema (date)

Utilizado para exibir ou alterar a data/hora do sistema.

Sintaxe:

- date

Exibir e Alterar o Calendário (cal)

Utilizado para exibir ou alterar o calendário.

Exemplos:

- cal
- cal 11 2014
- cal -3

Reiniciando o Sistema com reboot

Utilizado para reiniciar o computador.

Sintaxe:

- reboot

Desligando o Sistema com halt

Utilizado para desligar o computador.

Sintaxe:

- halt

Introdução

O que é Shell?

- É um programa que permite ao usuário interagir com o Sistema Operacional (SO) através dos **comandos** digitados no teclado.



Figura: Logo do Shell Script

Tipos de Shell

Tipos de Shell:

- *bash* (Bourne Again Shell).
 - *sh* (Bourne Shell).
 - *ksh* (Korn Shell).
 - *csh* (C Shell).
 - *zsh* (Z Shell)
-
- Como saber o shell que estou utilizando?
 - echo \$SHELL

Shell Script

O que é:

- Pode se referir a mais de uma linguagem.
- É uma linguagem de *script*, usada em vários SO's, com diferentes dialetos, dependendo do interpretador de comandos (shell) utilizado.

Shell Script

Identificador de códigos shell:

- Todo arquivo com códigos shell deve ter, na primeira linha, o identificador do interpretador a ser utilizado.
- A "linha mágica" que chama o interpretador *bash* é:
 - `#!/bin/bash`

Shell Script

Permissão de execução:

- Após adicionarmos o comando (**#/bin/bash**), devemos dar permissão de execução no arquivo.
- Para dar permissão de execução, utiliza-se o seguinte comando¹:
 - **\$ chmod +x nomeDoScript.sh**

¹É necessário estar logado como administrador

Shell Script

Execução:

- Para executar o arquivo `.sh`, faça:
 - `$./nomeDoScript.sh` ou
 - `$ bash nomeDoScript.sh`



Figura: `#!/bin/bash`

Variáveis em Shell Script

Definição de variável:

- O *bash*, assim como outras linguagens, possui o conceito de variável.
- Para definir uma variável²:
 - **nome_da_variavel="valor_da_variavel"**
- Para exibir o valor de uma variável:
 - **echo \$nome_da_variavel**

²não deve haver espaços

Condicionais

As estruturas condicionais deslocam o fluxo de execução, conforme determinadas condições.

- **if:**

```
if  
then  
    comandos  
elif  
    comandos  
else  
    comandos  
fi
```

Figura: Sintaxe da estrutura condicional if

Condicionais

As estruturas condicionais deslocam o fluxo de execução, conforme determinadas condições.

- if:

```
if [ 5 -lt 7 ]  
then  
    echo '5 é menor que 7'  
fi
```

Figura: Exemplo de uso do if

- -lt: *less than*
- -gt: *greater than*
- -le: *less equal*
- -ge: *greater equal*
- -eq: *equal*
- -ne: *not equal*

Prática 1: Criar um programa em shell script que verifica se um número é positivo ou negativo.

- Dicas:
 - Para ler um número usamos a palavra-chave: *read*

Prática 2: Criar um programa em shell script que verifica se um número é par ou ímpar.

- Dicas:
 - Para obter o resto da divisão de um número por outro, utilizamos o símbolo %.

Condicionais

As estruturas condicionais deslocam o fluxo de execução, conforme determinadas condições.

● case:

```
case in
  opcao1)
    comandos
    ;;
  opcao2)
    comandos
    ;;
  *)
    operação padrão
esac
```

Figura: Sintaxe da estrutura condicional case

Prática 3: Criar um programa em shell script que, a partir da leitura de um número, mostre na tela o número digitado. Considere que o programa lê os números 1, 2, 3 e 4, e qualquer outro número aparece a mensagem: **Você digitou outro número.**

- Dica: usar a estrutura condicional case.

Estrutura de repetição: for

Loops são estruturas que permitem fazer iterações sobre algum dado.

- for:

```
for <var> in  
do  
  
done  
</var>
```

Figura: Sintaxe da estrutura de repetição for

Estrutura de repetição: for

- for:

```
for <var> in  
do  
  
done  
</var>
```

Figura: Exemplo da sintaxe do for

Prática 4: Criar um programa em shell script que exibe a sequência dos números: 1, 2, 3, 4, 5, 6, 7, 8, 9 e 10.

- Dica: usar a estrutura condicional for.

Prática 5: Criar o mesmo programa proposto na prática4 usando o comando **seq**.

- Vamos fazer juntos?

- while:

```
while  
do  
  
done
```

Figura: Exemplo da sintaxe do while

Prática 6: Criar o mesmo programa da prática 4, em shell script, que exiba na tela os elementos de 1 a 10, usando a estrutura de repetição while.

- Dicas:
 - Use a estrutura de comparação -lt (less than), s

Considerações finais

- O mundo do Linux e dos shell scripts é muito mais amplo!
- Existem diversos outros comandos e diversas outras funcionalidades do Linux e de shell script.
- É possível ir muito mais além: podemos criar scripts profissionais que automatizam os serviços e ajudam muito no dia-a-dia.

Agradecimentos

- A Comissão Organizadora da VI SECITEC de Formosa, pela oportunidade.
- Aos participantes do minicurso. Até a próxima! =)

Contato com os autores:

Dúvidas/Sugestões/Críticas:

- Mário Lemes
email: mariolpu@gmail.com
- Uyara Ferreira
email: uyara.ifg@gmail.com