

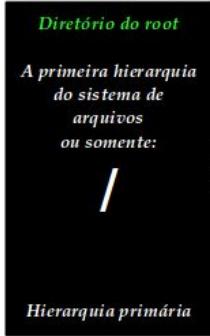


Linux

Terminal Linux

Fundamentos, Comandos e Bash Script

Curso: Ciência da Computação – Sistemas Operacionais
Professor: Fernando Costa Leite



/bin/	Binários principais dos usuários
/boot/	Arquivos do sistema de Boot
/dev/	Arquivos de dispositivos
/etc/	Arquivos de configuração do sistema
/home/	Diretório dos usuários comuns do sistema
/lib/	Bibliotecas essenciais do sistema e os módulos do kernel
/media/	Diretório de montagem de dispositivos
/mnt/	Diretório de montagem de dispositivos - <i>Mesmo que "media"</i>
/opt/	Instalação de programas não oficiais da distribuição ou por conta do usuário
/sbin/	Armazena arquivos executáveis que representam comandos administrativos. Exemplo: shutdown
/srv/	Diretório para dados de serviços fornecidos pelo sistema
/tmp/	Diretório para arquivos temporários
/usr/	Segunda hierarquia do sistema, onde ficam os usuários comuns do sistema e programas
/var/	Diretório com arquivos variáveis gerados pelos programas do sistema. Exemplo: logs, spool de impressoras, e-mail e cache
/root/	Diretório do usuário root – usuário root tem total poderes sobre o sistema, podendo instalar, desinstalar e configurá-lo.
/proc/	Diretório virtual controlado pelo Kernel com configuração total do sistema.

Terminal e Shell

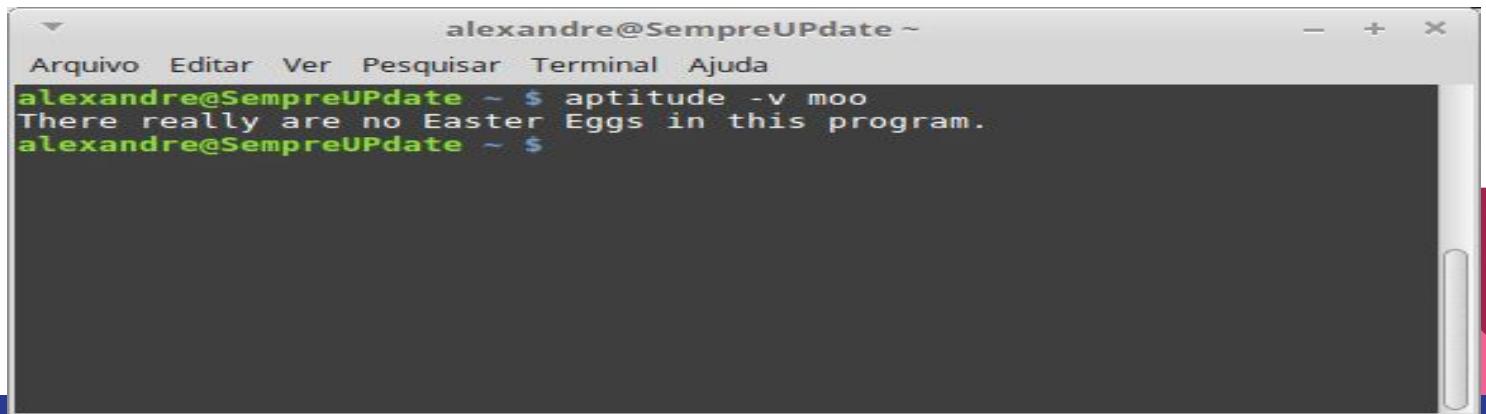
Shell: interface entre usuário e o kernel

Terminal: programa que exibe o shell

Principais shells: bash, zsh, fish

💬 Comando + argumento + opções

Exemplo:



```
alexandre@SempreUPdate ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
alexandre@SempreUPdate ~ $ aptitude -v moo
There really are no Easter Eggs in this program.
alexandre@SempreUPdate ~ $
```

Comandos de Navegação

Comando	Ação
pwd	Mostra o diretório atual
ls	Lista arquivos e pastas
cd	Muda de diretório
mkdir	Cria diretório
rmdir	Remove diretório
touch	Cria arquivo vazio

Manipulação de Arquivos

Comando	Ação
cp	Copiar arquivo
mv	Mover ou renomear
rm	Apagar arquivo
cat	Mostrar conteúdo
less	Ler por páginas
find / -name arquivo	Buscar arquivo

Permissões e Usuários

💡 Cada arquivo possui:

- **Usuário (owner)**
- **Grupo (group)**
- **Outros (others)**

Tipo	Símbolo	Valor
Leitura	r	4
Escrita	w	2
Execução	x	1

Gerenciamento de Processos

Comando	Ação
ps	Lista processos
top / htop	Monitora em tempo real
kill <PID>	Encerra processo
df -h	Uso de disco
free -h	Memória
uptime	Tempo ligado

Redirecionamentos e Pipes



Permite montar “mini pipelines” de comandos combinados.

Operador	Função
>	Redireciona saída (sobrescreve)
>>	Acrescenta
<	Redireciona entrada

bash

```
cat /etc/passwd | grep bash > usuarios.txt
```

Variáveis de Ambiente

Guardam informações do sistema e do usuário.

 Variáveis podem ser usadas dentro de scripts para automação.

```
bash

echo $HOME
echo $USER
export PATH=$PATH:/meu/script
```

Introdução ao Bash Script

Bash Script: sequência de comandos automatizados.

Início obrigatório: `#!/bin/bash`

Exemplo:

```
#!/bin/bash
echo "Olá, $USER!"
date
```

Estruturas de Controle - Condisional

```
read -p "Digite um número: " num
if [ $num -gt 10 ]; then
    echo "Maior que 10"
else
    echo "Menor ou igual a 10"
fi
```

Estruturas de Controle - Loop

```
for i in {1..5}; do  
    echo "Número $i"  
done
```

Funções e Reutilização

🧠 O que é uma função?

Uma função em Bash é um bloco de comandos reutilizável.

Permite organizar scripts e evitar repetição de código.

Pode receber parâmetros e retornar valores (via echo ou códigos de saída).

🧩 Parâmetros e retorno:

- `$1, $2, $3` → parâmetros passados à função
- `$@` → todos os parâmetros
- `return` → retorna um código numérico (0 = sucesso)

Funções e Reutilização



Escopo e retorno:

Variáveis dentro da função são globais por padrão

Para torná-las locais: usar modificador local

Funções e Reutilização - Exemplos

```
#!/bin/bash

verificar_numero() {
    if [ $1 -gt 10 ]; then
        echo "Maior que 10"
    else
        echo "Menor ou igual a 10"
    fi
}

verificar_numero 7
verificar_numero 15
```

```
#!/bin/bash

dizer_ola() {
    echo "Olá, $1!"
}
```

```
dizer_ola Fernando
dizer_ola Maria
```

Usuários, Grupos e Segurança

adduser, deluser, addgroup

su e sudo para troca de usuário

Arquivo /etc/sudoers controla permissões elevadas

 Separar contas evita erros e aumenta a segurança.



Gerenciamento de pacotes

Debian/Ubuntu: apt

Fedora/Red Hat: dnf, yum

Multi-distro: snap

Gerenciamento e Serviços

Gerenciador de inicialização: systemd

```
bash  
  
systemctl start nginx  
systemctl stop nginx  
systemctl status nginx
```

Comandos de Rede

Comando	Uso
ifconfig, ip addr	Exibe interfaces
ping	Testa conectividade
netstat, ss	Verifica portas
curl, wget	Baixa conteúdo web

Ajuda e Documentação no Linux

Comando	Descrição
man comando	manual detalhado
comando --help	opções rápidas
whatis comando	descrição curta
which comando	mostra o comando
tldr comando	Exemplos resumidos

Desafios Práticos

- ① Crie uma pasta projeto e 3 arquivos .txt dentro dela.
- ② Liste todos os arquivos que contêm uma palavra específica com grep.
- ③ Crie um script que receba um nome e exiba “Olá, [nome]”.
- ④ Crie um script que mostre os 5 processos com maior uso de CPU.
- ⑤ Altere as permissões de um arquivo para execução apenas pelo dono.

FIM!

