

Curso Básico de Linux para Iniciantes

Com ilustrações e respostas aos exercícios

Produzido por OpenAI - 2025

Sumário

1. 1. Modulo 1: Introducao ao Linux
2. 2. Modulo 2: Instalando o Linux
3. 3. Modulo 3: Navegando no Sistema
4. 4. Modulo 4: Comandos Basicos no Terminal
5. 5. Modulo 5: Permissoes e Usuarios
6. 6. Modulo 6: Gerenciamento de Pacotes
7. 7. Modulo 7: Editores de Texto no Terminal
8. 8. Modulo 8: Processos e Servicos
9. 9. Modulo 9: Redes no Linux
10. 10. Modulo 10: Dicas e Boas Praticas
11. 11. Exercícios e Respostas
12. 12. Projeto Final

1. Modulo 1: Introducao ao Linux

- - O que é Linux?
- - Historia do Linux e do Software Livre
- - Distribuicoes Linux populares (Ubuntu, Fedora, Debian, Mint)
- - Diferencas entre Linux e outros sistemas (Windows/macOS)
- - Conceitos de Kernel, Shell e Terminal

? O que é Linux?

Linux é um sistema operacional baseado em Unix, que atua como uma interface entre o hardware do computador e os programas que você executa. Ele é gratuito, de código aberto e amplamente utilizado em servidores, dispositivos móveis (como Android), supercomputadores e também em desktops.

Ao contrário de sistemas operacionais proprietários, como o Windows ou o macOS, o Linux é desenvolvido de forma colaborativa por comunidades de desenvolvedores do mundo todo. O sistema é conhecido por sua estabilidade, segurança e flexibilidade, podendo ser personalizado para diversas finalidades.



? História do Linux e do Software Livre

Linux:

O Linux foi criado em 1991 por Linus Torvalds, um estudante de ciência da computação na Finlândia. Inicialmente, ele queria apenas criar um núcleo (kernel) de sistema operacional como hobby. Ao publicar o projeto na internet, milhares de desenvolvedores começaram a colaborar com o código, ajudando a transformá-lo em um sistema completo.

Software Livre:

O conceito de software livre surgiu antes do Linux, com o movimento liderado por Richard Stallman, fundador da Free Software Foundation (FSF), em meados dos anos 1980. O objetivo do movimento era garantir que os usuários tivessem liberdade de usar, estudar, modificar e compartilhar programas.

O sistema operacional GNU foi iniciado por Stallman, e o kernel Linux foi o elemento que o tornou funcional, formando o que muitos chamam de GNU/Linux.

? Distribuições Linux Populares

O Linux não é distribuído como um único pacote. Em vez disso, existem centenas de distribuições (distros) que incluem o kernel Linux mais um conjunto de aplicativos, gerenciadores de pacotes, interfaces gráficas e configurações.

Algumas das mais populares são:

- **Ubuntu:** Fácil de usar, ideal para iniciantes, com ampla comunidade e suporte.
- **Fedora:** Patrocinada pela Red Hat, tem foco em tecnologias recentes e estabilidade.
- **Debian:** Uma das distros mais antigas e estáveis. Muitas outras distribuições são baseadas nela.
- **Linux Mint:** Baseada no Ubuntu, com aparência familiar para usuários do Windows.

Essas distribuições diferem na aparência, ferramentas padrão e métodos de gerenciamento, mas todas compartilham o núcleo Linux.

? Diferenças entre Linux e outros sistemas (Windows/macOS)

Característica	Linux	Windows	macOS
Licença	Livre e de código aberto	Proprietário	Proprietário

Característica	Linux	Windows	macOS
Personalização	Alta	Limitada	Limitada
Linha de comando	Muito utilizada e poderosa	Opcional, menos usada	Opcional
Instalação de software	Gerenciadores de pacotes	Arquivos .exe/.msi	App Store e arquivos .dmg
Segurança	Muito segura, menos vírus	Vulnerável, requer antivírus	Muito segura
Custo	Gratuito	Pago (normalmente embutido)	Pago (embutido no hardware)

O Linux se destaca em servidores, desenvolvedores e usuários que buscam controle e liberdade sobre o sistema. Já o Windows e o macOS tendem a focar mais em usuários finais com interface simplificada.

🔗 Conceitos de Kernel, Shell e Terminal

🔗 Kernel:

É o núcleo do sistema operacional, responsável por gerenciar recursos como memória, processos, arquivos e dispositivos. Ele faz a ponte entre o hardware e o software.

Exemplo: Quando você abre um arquivo, é o kernel que conversa com o disco para buscar esse conteúdo.

🔗 Shell:

É o interpretador de comandos que permite interagir com o sistema. Pode ser uma interface de texto (como o Bash) ou gráfica.

Exemplo: Quando você digita `ls`, o shell interpreta isso como "listar arquivos".

🔗 Terminal:

É o programa que permite você digitar comandos para o shell. Ele fornece uma interface textual onde você pode executar tarefas administrativas, rodar scripts, instalar programas etc.

Exemplo: No Ubuntu, o terminal pode ser aberto pressionando `Ctrl + Alt + T`.

Deseja que eu insira esse conteúdo no seu curso e gere uma nova versão do PDF ou DOCX com esse texto incluído?

2. Modulo 2: Instalando o Linux

- - Escolhendo uma distribuicao
- - Criando um pendrive bootavel (Rufus, Etcher)
- - Dual boot vs Maquina Virtual
- - Instalacao do Ubuntu passo a passo
- - Primeiro boot e interface grafica (GNOME/KDE)

? Escolhendo uma Distribuição Linux

Antes de instalar o Linux, é importante escolher uma distribuição (ou "distro") que atenda suas necessidades. Cada distribuição tem características específicas:

Distribuição	Ideal para	Destaques
Ubuntu	Iniciantes	Interface amigável, boa compatibilidade
Linux Mint	Usuários de Windows	Interface tradicional, leve e intuitiva
Fedora	Desenvolvedores	Tecnologias recentes, estabilidade
Debian	Avançados/Servidores	Extremamente estável, base para outras
Arch Linux	Usuários avançados	Total controle, instalação manual

➡ Para iniciantes, recomenda-se Ubuntu ou Linux Mint por sua facilidade de uso, ampla documentação e grande comunidade.

? Criando um Pendrive Bootável

Para instalar o Linux, você precisa criar um pendrive bootável com a imagem da distribuição (arquivo .ISO).

🎱 Ferramentas populares:

✓ Rufus (Windows)

- Baixe o [Rufus](#).

- Insira o pendrive.
- Escolha a ISO do Linux.
- Clique em “Iniciar”.
- Aguarde até o processo finalizar.

✓ *Etcher (Windows, macOS, Linux)*

- Baixe o [balenaEtcher](#).
- Interface simples: selecione ISO, selecione o pendrive, clique em “Flash!”.
- Funciona em qualquer sistema operacional.

Dica: Use um pendrive de pelo menos 4 GB (preferencialmente 8 GB ou mais).

🔗 Dual Boot vs Máquina Virtual

穿 Dual Boot

Instala dois sistemas operacionais no mesmo computador (por exemplo, Windows e Linux). Na inicialização, você escolhe qual sistema usar.

Vantagens:

- Melhor desempenho.
- Acesso direto ao hardware.

Desvantagens:

- Risco de sobrescrever partições.
- Requer mais planejamento.

離 Máquina Virtual

Executa o Linux dentro do Windows/macOS usando softwares como VirtualBox ou VMware.

Vantagens:

- Sem risco de alterar o sistema principal.
- Fácil de instalar e remover.

Desvantagens:

- Desempenho limitado.
- Acesso reduzido ao hardware (gráficos, USB etc).

Recomendação:

Para testes ou aprendizado, use máquina virtual. Para uso mais sério, opte por dual boot ou instalação única.

Instalação do Ubuntu Passo a Passo

1. Baixe a ISO do Ubuntu em ubuntu.com/download.
 2. Crie um pendrive bootável com Rufus ou Etcher.
 3. Reinicie o computador e entre na BIOS/UEFI (geralmente tecla F2, F10, DEL ou ESC).
 4. Configure o boot pelo pendrive.
 5. Selecione "Try Ubuntu" ou "Install Ubuntu".
 6. Escolha o idioma e o layout do teclado.
 7. Escolha "Instalar Ubuntu ao lado do Windows" ou "Apagar disco" (atenção: esta última apaga tudo!).
 8. Configure seu fuso horário.
 9. Crie o nome de usuário e senha.
 10. Clique em "Instalar agora" e aguarde.
 11. Após finalização, remova o pendrive e reinicie o computador.
-

? Primeiro Boot e Interface Gráfica (GNOME/KDE)

? Primeira inicialização:

Após o primeiro boot, você verá a interface gráfica do Ubuntu (por padrão, GNOME). Faça login com seu usuário e explore o ambiente.

付 GNOME:

- Interface padrão no Ubuntu.
- Foco na simplicidade e produtividade.
- Painel superior com atalhos e menu de aplicativos.

里 KDE Plasma:

- Visual moderno e altamente personalizável.
- Repleto de recursos, ideal para quem gosta de customizar tudo.
- Pode ser instalado em distribuições como Kubuntu.

Você pode trocar o ambiente gráfico mesmo depois de instalar, mas é melhor escolher na instalação inicial.

Dica para iniciantes: Fique com o GNOME, é simples, estável e bem suportado.

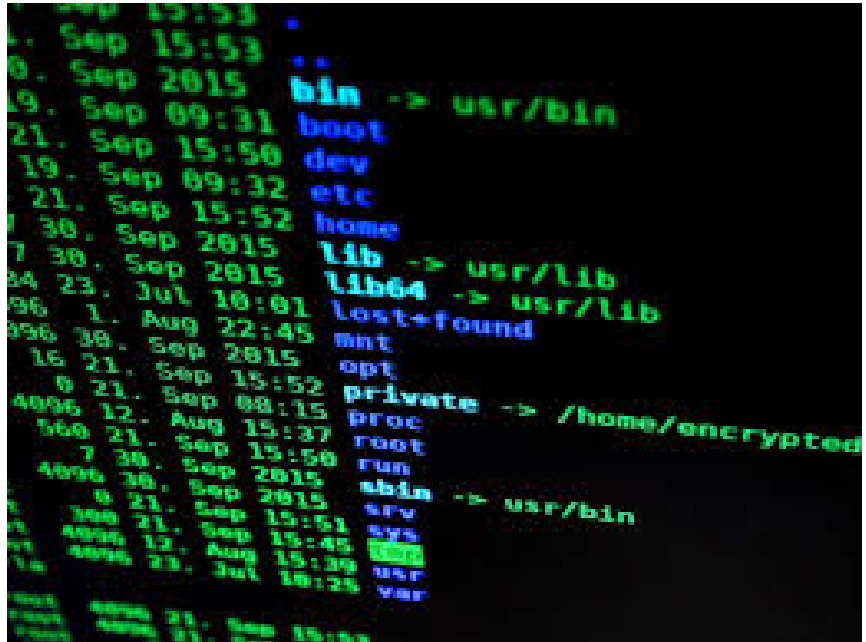


3. Modulo 3: Navegando no Sistema

- - Estrutura de diretorios do Linux (/ , /home , /etc , /bin , /usr , etc.)
- - Tipos de arquivos no Linux
- - Conceito de permissões (leitura, escrita, execucao)
- - Interface grafica vs Terminal

🔍 Estrutura de Diretórios do Linux

O Linux segue uma estrutura de diretórios hierárquica, semelhante a uma árvore. Tudo começa pela raiz (/), e a partir dela, se ramificam os outros diretórios.



Diretório	Função Principal
/	Raiz do sistema, base de todos os diretórios.
/home	Onde ficam os arquivos dos usuários comuns (ex: /home/joao).
/etc	Arquivos de configuração do sistema e serviços.
/bin	Programas essenciais do sistema (binários executáveis, como ls, cp).
/usr	Aplicações e arquivos de usuários. Possui subpastas como /usr/bin.
/var	Arquivos variáveis, como logs, emails e cache.
/tmp	Arquivos temporários. São apagados após reinicializações.
/root	Diretório pessoal do usuário root (administrador).
/dev	Arquivos que representam dispositivos (discos, impressoras etc).

Diretório

Função Principal

/mnt ou /media Pontos de montagem para dispositivos externos.

撤 Exemplo: O comando `cd /etc` leva você para a pasta com arquivos de configuração do sistema.

? Tipos de Arquivos no Linux

No Linux, nem tudo é apenas "arquivo comum". O sistema lida com diferentes tipos de arquivos, cada um com função específica:

Tipo	Descrição	Exemplo
Arquivo regular	Textos, scripts, imagens, binários	documento.txt
Diretório	Uma pasta que pode conter arquivos	/home, /etc
Link simbólico	Atalho para outro arquivo	meuarquivo -> /outro/arquivo
Dispositivo	Representação de hardware	/dev/sda1
FIFO (named pipe)	Comunicação entre processos	usado por serviços do sistema
Socket	Comunicação de rede entre programas	/run/docker.sock

Use `ls -l` para ver o tipo do arquivo. Exemplo: `d` para diretório, `-` para arquivo comum, `l` para link simbólico.

? Conceito de Permissões (Leitura, Escrita, Execução)

O Linux é um sistema multiusuário, por isso o controle de permissões é fundamental para a segurança.

Cada arquivo ou pasta tem permissões para três tipos de usuários:

- **Usuário (owner)** – Quem criou o arquivo.
- **Grupo (group)** – Grupo ao qual o arquivo pertence.
- **Outros (others)** – Todos os demais usuários.

茶 Tipos de permissões:

Permissão	Símbolo	Função
Leitura	r	Ver conteúdo do arquivo ou listar pasta
Escrita	w	Modificar arquivo ou criar arquivos em uma pasta
Execução	x	Executar programas ou acessar uma pasta

% Exemplo com ls -l:

```
-rwxr-xr-- 1 joao grupo 1234 mai 10 12:00 script.sh
```

- rwx → dono (joao) pode ler, escrever e executar.
- r-x → grupo pode ler e executar.
- r-- → outros podem apenas ler.

Use o comando `chmod` para alterar permissões e `chown` para mudar o dono de um arquivo.

? Interface Gráfica vs Terminal

† Interface Gráfica (GUI – Graphical User Interface)

- Oferece uma experiência visual com janelas, ícones e menus.
- Exemplos: GNOME, KDE, XFCE.
- Ideal para iniciantes e tarefas cotidianas.

泉 Terminal (CLI – Command Line Interface)

- Acesso direto ao sistema por comandos digitados.
- Mais rápido e poderoso para tarefas administrativas.
- Indispensável para servidores, automações, programação e instalação de pacotes.

Comparativo	Interface Gráfica	Terminal
Curva de aprendizado	Baixa	Mais alta (no início)
Flexibilidade	Limitada	Muito alta
Velocidade	Depende da máquina	Muito rápida
Recursos do sistema	Alta	Baixa (ideal para PCs fracos)

Dica: No Ubuntu, pressione Ctrl + Alt + T para abrir o terminal.

4. Modulo 4: Comandos Basicos no Terminal

- - Abrindo o terminal
 - - Comandos de navegacao: pwd, cd, ls, tree
 - - Manipulacao de arquivos e diretorios: touch, mkdir, cp, mv, rm
 - - Visualizando conteudo: cat, less, head, tail
 - - Buscas: find, grep
-

Abrindo o Terminal

O terminal é a porta de entrada para comandos no Linux. Através dele, você pode navegar pelo sistema, criar arquivos, instalar programas, e muito mais.

Como abrir:

- **Ubuntu: Pressione `Ctrl + Alt + T`.**
- **Também pode procurar por "Terminal" no menu de aplicativos.**
- **Em máquinas virtuais, geralmente já há um atalho fixado.**

Comandos de Navegação

Esses comandos ajudam você a se localizar e mover entre diretórios:

Comando	Função	Exemplo
pwd	Mostra o caminho do diretório atual	pwd → /home/usuario
cd	Muda para outro diretório	cd /etc
cd ..	Sobe um nível na hierarquia	cd ..
ls	Lista arquivos e pastas do diretório atual	ls -l (detalhado), ls -a (ocultos)
tree	Mostra estrutura em formato de árvore	tree (pode precisar instalar)

% Dica: Use cd ~ para voltar ao seu diretório pessoal.

? Manipulação de Arquivos e Diretórios

Aqui estão os principais comandos para criar, copiar, mover e excluir arquivos e pastas:

Comando	Função	Exemplo
touch	Cria um novo arquivo vazio	touch texto.txt
mkdir	Cria um diretório	mkdir pasta1
cp	Copia arquivos ou pastas	cp a.txt b.txt ou cp -r pasta1/ pasta2/
mv	Move ou renomeia arquivos	mv a.txt documentos/ ou mv a.txt novo.txt
rm	Remove arquivos ou pastas	rm arquivo.txt ou rm -r pasta1/

⚠ Atenção: Use rm com cuidado! Ele não vai para a "Lixeira".

? Visualizando Conteúdo de Arquivos

Esses comandos ajudam a ler o conteúdo de arquivos de texto diretamente no terminal:

Comando	Função	Exemplo
cat	Mostra todo o conteúdo do arquivo	cat texto.txt
less	Abre arquivo para leitura paginada	less texto.txt (pressione q para sair)

Comando	Função	Exemplo
head	Mostra as 10 primeiras linhas	head log.txt ou head -n 5 log.txt
tail	Mostra as 10 últimas linhas	tail log.txt ou tail -n 20 log.txt
% less é ótimo para arquivos grandes porque você pode rolar com as setas.		

? Buscas com find e grep

% find – Localiza arquivos e pastas no sistema:

```
find /caminho -name "nome"
```

Exemplos:

- `find . -name "*.txt"` → Encontra todos os arquivos .txt na pasta atual.
- `find / -name "arquivo.txt"` → Procura por "arquivo.txt" no sistema (requer sudo).

% grep – Busca conteúdo dentro de arquivos:

```
grep "palavra" arquivo.txt
```

Exemplos:

- `grep "erro" log.txt` → Mostra linhas do arquivo que contenham "erro".
- `grep -r "senha" /etc/` → Busca "senha" recursivamente em /etc.

責 Combine comandos: `cat arquivo.txt | grep "Linux"`

✓ Agora que você conhece esses comandos básicos, já pode realizar muitas tarefas no Linux **sem sair do terminal!**

5. Modulo 5: Permissoes e Usuarios

- - Entendendo chmod, chown, chgrp
- - Conceito de root e sudo
- - Criando e gerenciando usuarios: adduser, passwd, usermod, deluser
- - Gerenciando grupos

? Entendendo chmod, chown e chgrp

% chmod - Altera permissões de arquivos ou diretórios

Permissões são representadas por letras ou números. O comando chmod permite conceder ou retirar essas permissões.

⇒ Sintaxe com letras:

```
chmod [u|g|o]+[r|w|x] arquivo
```

Símbolo	Significado
u	usuário (owner)
g	grupo
o	outros
+	adiciona permissão
-	remove permissão

Exemplos:

```
chmod u+x script.sh # Adiciona permissão de execução para o dono  
chmod go-w arquivo.txt # Remove escrita de grupo e outros
```

? Sintaxe com números:

```
chmod 755 arquivo.sh
```

Dígito	Permissões
7	rwX (leitura, escrita, execução)
6	rw-
5	r-X
4	r--

Exemplo:

```
chmod 644 texto.txt    # Dono: leitura e escrita | Grupo e outros:
leitura
```

chown - Muda o dono de um arquivo ou diretório

```
chown novo_dono arquivo.txt
chown joao:grupo arquivo.txt
```


chgrp - Muda apenas o grupo de um arquivo

```
chgrp grupo arquivo.txt
```

Conceito de root e sudo

root:

É o superusuário, com permissão total no sistema. Pode editar qualquer arquivo, instalar pacotes, excluir pastas do sistema etc.

 **Cuidado:** erros como root podem comprometer o sistema.

sudo:

Permite executar comandos de root temporariamente, com segurança e registro.

Exemplo:

```
sudo apt update
sudo rm -rf /diretorio/
```

O usuário precisa estar autorizado no grupo sudo para usar esse comando.

? Criando e Gerenciando Usuários

+ adduser - Cria um novo usuário

```
sudo adduser maria
```

- Cria a pasta /home/maria
 - Define senha
 - Solicita informações opcionais
-

% passwd - Altera a senha de um usuário

```
passwd maria
```

⇒ usermod - Modifica usuário existente

```
usermod -aG sudo maria # Adiciona ao grupo sudo
usermod -s /bin/bash maria # Altera o shell padrão
```

✕ deluser - Remove um usuário

```
sudo deluser maria
```

Para remover também a pasta /home:

```
sudo deluser --remove-home maria
```

? Gerenciando Grupos

Grupos ajudam a organizar permissões para múltiplos usuários.

+ Criar um grupo:

```
sudo addgroup desenvolvedores
```

Adicionar usuário a um grupo:

```
sudo usermod -aG desenvolvedores joao
```

Remover grupo:

```
sudo delgroup desenvolvedores
```

Ver grupos de um usuário:

```
groups joao
```

✓ Com esses comandos, você pode controlar o acesso ao sistema e arquivos de forma segura e eficaz!

6. Modulo 6: Gerenciamento de Pacotes

- - O que são pacotes e repositórios
- - APT (Ubuntu/Debian): apt update, apt install, apt remove
- - Outros gerenciadores: dnf, yum, pacman, snap, flatpak



? Módulo 6: Gerenciamento de Pacotes no Linux

廳 O que são pacotes e repositórios

- **Pacotes são arquivos que contêm programas ou bibliotecas necessárias para o funcionamento de um software no Linux. Um pacote pode conter:**
 - O código binário do programa
 - Scripts de instalação
 - Dependências (outros pacotes necessários)
- **Repositórios são servidores online que armazenam e distribuem pacotes. Cada distribuição Linux possui repositórios oficiais e pode incluir repositórios de terceiros.**

易 Exemplo:

Ao instalar o Firefox no Ubuntu, o sistema baixa o pacote .deb do repositório oficial, instala e configura automaticamente.

蘭 APT (Advanced Packaging Tool) – Usado no Ubuntu/Debian

疊 Atualizar a lista de pacotes:

```
sudo apt update
```

Verifica as versões mais recentes nos repositórios.

↓ *Instalar um pacote:*

```
sudo apt install nome-do-pacote
```

Exemplo: `sudo apt install vlc`

✕ *Remover um pacote:*

```
sudo apt remove nome-do-pacote
```

Exemplo: `sudo apt remove gimp`

粒 *Limpar pacotes não usados:*

```
sudo apt autoremove
```

里 *Outros gerenciadores de pacotes*

Cada distribuição possui seu próprio gerenciador:

Distribuição	Gerenciador	Comandos Básicos
Fedora, RHEL, CentOS	dnf	<code>sudo dnf install</code>
CentOS antigos	yum	<code>sudo yum install</code>
Arch Linux	pacman	<code>sudo pacman -S</code>
Qualquer distro	snap	<code>sudo snap install</code>
Qualquer distro	flatpak	<code>flatpak install</code>

? *Exemplos práticos*

Fedora - DNF:

```
sudo dnf update
sudo dnf install firefox
```

Arch - Pacman:

```
sudo pacman -Syu    # Atualiza o sistema
```

```
sudo pacman -S vlc
```

Snap:

```
sudo snap install code --classic
```

Flatpak:

```
flatpak install flathub org.gimp.GIMP
```

Dicas importantes:

- Não misture tipos de pacotes (ex: Snap e APT) sem necessidade.
- Flatpak e Snap são bons para versões recentes de apps.
- Sempre atualize antes de instalar: `sudo apt update`

✓ Compreender os pacotes e os gerenciadores é essencial para instalar, atualizar e manter seu sistema Linux funcional e seguro.

7. Modulo 7: Editores de Texto no Terminal

- - nano simples e facil
- - vim comandos basicos para iniciantes
- - Salvando e saindo com seguranca

? Módulo 7: Editores de Texto no Terminal Linux

nano: simples e fácil para iniciantes

O nano é um editor de texto leve, intuitivo e fácil de usar diretamente no terminal. Ideal para quem está começando no Linux.

Abrindo um arquivo com nano:

```
nano arquivo.txt
```

Se o arquivo não existir, o nano irá criá-lo automaticamente.

Comandos básicos do nano:

- **CTRL + O** → Salvar o arquivo ("Write Out")
- **CTRL + X** → Sair do editor
- **CTRL + K** → Cortar uma linha
- **CTRL + U** → Colar a linha
- **CTRL + W** → Procurar texto
- **CTRL + G** → Ajuda com todos os comandos

△ Os comandos aparecem na parte inferior da tela. O símbolo ^ significa **CTRL** (ex: ^X = CTRL + X).

vim: poderoso, mas exige prática

O vim é um editor muito avançado e amplamente usado por desenvolvedores e administradores de sistemas. É mais difícil no início, mas muito eficiente com prática.

撤 Abrindo um arquivo com vim:

```
vim arquivo.txt
```

磋 Modos do vim:

- **Modo Normal (inicial):** navegação e comandos
- **Modo de Inserção:** escrever texto
- **Modo de Comando:** salvar, sair, buscar

蘭 Comandos básicos:

- i → Entrar no **modo de inserção**
 - ESC → Voltar ao **modo normal**
 - :w → Salvar
 - :q → Sair
 - :wq → Salvar e sair
 - :q! → Sair sem salvar
 - u → Desfazer
 - /palavra → Buscar palavra
 - dd → Deletar linha
-

穿 Salvando e saindo com segurança

? Em nano:

- Pressione CTRL + O → Enter para salvar
- Pressione CTRL + X para sair

? Em vim:

- Pressione ESC para sair do modo de inserção
- Digite:
 - :w para salvar
 - :q para sair
 - :wq para salvar e sair

- `:q!` para sair sem salvar

撤 Dica:

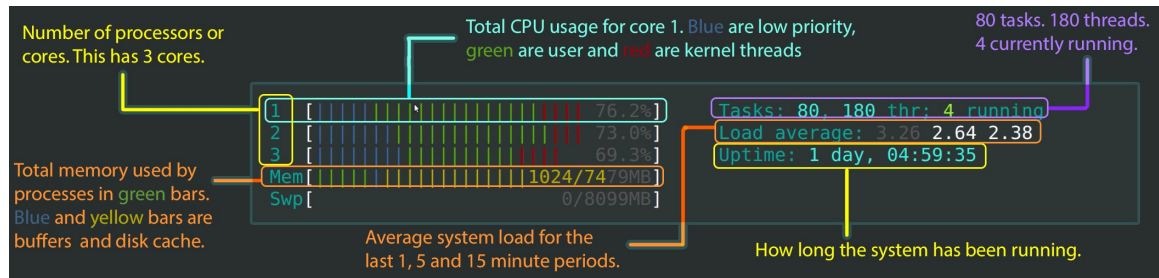
Para edições rápidas e simples, use nano.

Para tarefas mais avançadas (edição de código, macros, plugins), vim é mais indicado.

✓ Agora você pode criar, editar e salvar arquivos de texto no Linux com segurança, usando tanto o simples nano quanto o poderoso vim.

8. Modulo 8: Processos e Servicos

- - Monitorando o sistema: top, htop, ps, kill
- - Servicos e systemd: systemctl start/stop/status
- - Agendando tarefas: cron e crontab



? Módulo 8: Monitoramento e Gerenciamento no Linux

🗉 Monitorando o sistema

- **top**
Mostra os processos ativos em tempo real, uso de CPU, memória, tempo de atividade e mais.

top

- Use q para sair.
- **htop**
Versão mais amigável e colorida do top. Precisa estar instalado:

```
sudo apt install htop
```

htop

- Navegação com setas, F10 para sair.
- **ps**
Exibe processos em execução. Exemplo para ver processos do usuário atual:

```
ps aux
```

- Para encontrar um processo específico:

```
ps aux | grep nome_do_processo
```

- **kill**

Envia sinal para terminar um processo. Use com o PID (Process ID).

```
kill PID
```

- Para forçar:

```
kill -9 PID
```

⚙️ Serviços e systemd

O systemd é o sistema init padrão na maioria das distribuições modernas. Ele gerencia serviços (daemons), que são programas que rodam em segundo plano.

- **Iniciar um serviço**

```
sudo systemctl start nome_do_serviço
```

- **Parar um serviço**

```
sudo systemctl stop nome_do_serviço
```

- **Ver status do serviço**

```
sudo systemctl status nome_do_serviço
```

- **Habilitar serviço para iniciar automaticamente na inicialização**

```
sudo systemctl enable nome_do_serviço
```

- **Desabilitar serviço na inicialização**

```
sudo systemctl disable nome_do_serviço
```

🕒 Agendando tarefas com cron e crontab

- **cron é um serviço que executa comandos em horários programados.**
- **crontab é o arquivo onde se define essas tarefas.**

Comandos básicos:

- Editar o crontab do usuário atual:

```
crontab -e
```

- Listar tarefas agendadas:

```
crontab -l
```

- Remover todas as tarefas:

```
crontab -r
```

Formato de uma linha no crontab:

```
* * * * * comando
```

Os asteriscos representam:

- Minuto (0-59)
- Hora (0-23)
- Dia do mês (1-31)
- Mês (1-12)
- Dia da semana (0-7) (domingo é 0 ou 7)

Exemplo: Executar script.sh todo dia às 3 da manhã

```
0 3 * * * /home/usuario/script.sh
```

✓ Com esses comandos, você pode monitorar o uso do sistema, controlar serviços essenciais e agendar tarefas automáticas no Linux.

9. Módulo 9: Redes no Linux

- - Verificando conexões: ip a, ping, netstat, ss
 - - Configurando IP manualmente
 - - Usando ssh para acessar máquinas remotamente
-

? Módulo 9: Redes e Acesso Remoto no Linux

? Verificando conexões

- **ip a (ou ip addr)**

Exibe as interfaces de rede e seus endereços IP configurados.

```
ip a
```

- **ping**

Testa a conectividade com outro host na rede ou internet.

```
ping google.com
```

- Para parar o ping, pressione CTRL + C.

- **netstat**

Mostra conexões de rede, tabelas de roteamento e estatísticas. (Pode precisar instalar)

```
netstat -tuln
```

- -t TCP, -u UDP, -l só serviços ouvindo, -n números em vez de nomes.

- **ss (substituto moderno do netstat)**

Exibe sockets de rede com detalhes.

```
ss -tuln
```

⚙ Configurando IP manualmente

Para configurar um IP estático temporariamente em uma interface (exemplo eth0):

```
sudo ip addr add 192.168.1.100/24 dev eth0
```

```
sudo ip link set eth0 up
```

```
sudo ip route add default via 192.168.1.1
```

- 192.168.1.100/24: endereço IP e máscara.
- 192.168.1.1: gateway padrão.

Para configurar permanentemente, o método varia conforme a distribuição e gerenciador de rede (NetworkManager, netplan, etc). Exemplo básico para Ubuntu com netplan:

```
network:
  version: 2
  ethernets:
    eth0:
      addresses: [192.168.1.100/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Salve no arquivo /etc/netplan/01-netcfg.yaml e aplique:

```
sudo netplan apply
```

Usando SSH para acessar máquinas remotamente

- **SSH (Secure Shell)** permite acessar outro computador Linux pela rede, de forma segura.
- **Conectar a um servidor remoto:**

```
ssh usuario@ip_do_servidor
```

Exemplo:

```
ssh joao@192.168.1.50
```

- Na primeira conexão, confirme a chave do host digitando yes.
- Informe a senha do usuário remoto para autenticação.
- **Copiar arquivos via SCP:**

```
scp arquivo.txt usuario@ip_do_servidor:/caminho/destino/
```

- Gerar par de chaves SSH para login sem senha:

```
ssh-keygen
ssh-copy-id usuario@ip_do_servidor
```

✓ Com esses conhecimentos, você pode diagnosticar problemas de rede, configurar IPs e acessar outros computadores com segurança.

10. Modulo 10: Dicas e Boas Praticas

- - Comandos perigosos a evitar
- - Como buscar ajuda: man, --help, foruns, Stack Overflow
- - Onde continuar aprendendo



? Módulo 10: Boas Práticas e Recursos para Aprender Linux

⚠ Comandos perigosos a evitar

Alguns comandos no Linux podem causar danos graves ao sistema, apagar arquivos importantes ou deixar o sistema inutilizável se usados incorretamente. É fundamental saber quando e como usá-los.

- `rm -rf /`
Apaga recursivamente todos os arquivos a partir da raiz, destruindo o sistema.
- `mkfs`
Formata uma partição, apagando todos os dados nela.
- `:(){ :|:& };:` (Fork Bomb)
Ataque que cria processos infinitamente, travando o sistema.
- `dd`
Se mal usado, pode sobrescrever discos inteiros, causando perda total de dados.
- Comandos com `sudo` devem ser usados com cuidado, pois têm permissões elevadas.

Dica: Sempre revise o comando antes de executar e faça backups!

? Como buscar ajuda

- **man comando**

Mostra o manual detalhado do comando. Exemplo:

```
man ls
```

- **comando --help**

Exibe um resumo rápido das opções do comando. Exemplo:

```
ls --help
```

- **Fóruns e comunidades online**
 - [Stack Overflow](#)
 - [Ask Ubuntu](#)
 - [LinuxQuestions.org](#)
 - Fóruns específicos da distribuição (ex: Ubuntu Forums)
 - **Documentação oficial e blogs**

Muitas distribuições possuem documentação oficial bem completa.
 - **Cursos e vídeos no YouTube**

Existem diversos conteúdos para todos os níveis.
-

📖 Onde continuar aprendendo

- **Documentação oficial do Linux**
<https://www.kernel.org/doc/html/latest/>
- **Guias de distribuições**
 - Ubuntu: <https://help.ubuntu.com/>
 - Fedora: <https://docs.fedoraproject.org/>
- **Livros**
 - “The Linux Command Line” por William Shotts
 - “Linux Pocket Guide” por Daniel J. Barrett
- **Plataformas de cursos online**
 - Udemy
 - Coursera
 - edX
 - Alura (Português)

- **Prática constante**

Nada substitui o uso diário e a experimentação.

✓ Seguindo essas práticas, você manterá seu sistema seguro e terá sempre fontes confiáveis para aprofundar seu conhecimento em Linux.

Exercícios com Respostas

Módulo 1 - Exercício

Pesquise as principais distribuicoes Linux e escolha uma para testar.

Resposta:

Resposta: Ubuntu e uma boa escolha para iniciantes. Baixe a ISO no site oficial.

Módulo 2 - Exercício

Crie um pendrive bootavel e tente iniciar o sistema em modo live.

Resposta:

Resposta: Use o programa Rufus no Windows para gravar a ISO em um pendrive.

Módulo 3 - Exercício

Abra o terminal e navegue ate seu diretorio pessoal usando comandos.

Resposta:

Resposta: Use 'cd ~' ou apenas 'cd' para ir ao diretorio pessoal.

Módulo 4 - Exercício

Crie uma pasta, um arquivo, e mova-os com comandos no terminal.

Resposta:

Resposta: mkdir teste; touch teste/arquivo.txt; mv teste/arquivo.txt ./

Módulo 5 - Exercício

Crie um novo usuario e altere as permissoes de um arquivo para ele.

Resposta:

Resposta: sudo adduser aluno; sudo chown aluno:aluno arquivo.txt

Módulo 6 - Exercício

Instale o programa 'htop' usando o gerenciador de pacotes.

Resposta:

Resposta: sudo apt update && sudo apt install htop

Módulo 7 - Exercício

Edite um arquivo de texto usando o nano e o vim.

Resposta:

Resposta: nano arquivo.txt; vim arquivo.txt; use :wq para salvar e sair do vim.

Módulo 8 - Exercício

Use o comando 'top' para encontrar o processo que mais consome CPU.

Resposta:

Resposta: top mostra os processos em tempo real, o primeiro geralmente é o que mais consome.

Módulo 9 - Exercício

Use o comando 'ping' para verificar a conexão com www.google.com.

Resposta:

Resposta: ping www.google.com deve retornar pacotes se estiver conectado.

Módulo 10 - Exercício

Use 'man ls' para entender todas as opções do comando ls.

Resposta:

Resposta: man ls abre o manual, use seta para baixo e 'q' para sair.

Projeto Final

Crie um script em Bash chamado 'organiza.sh' que:

- Crie pastas como 'Imagens', 'Documentos', 'Outros'.
- Mova arquivos da pasta Downloads para as respectivas pastas, com base na extensao.
- Gere um arquivo de log com as acoes realizadas.
- (Opcional) Agende a execucao automatica com crontab.

Exemplo de script:

```
#!/bin/bash
mkdir -p Imagens Documentos Outros
mv ~/Downloads/*.jpg Imagens/ 2>/dev/null
mv ~/Downloads/*.pdf Documentos/ 2>/dev/null
mv ~/Downloads/* Outros/ 2>/dev/null
echo "Arquivos organizados em $(date)" >> log.txt
```