

## Minicurso de Introdução ao GNU/Linux

IEEE Computer Society UnB

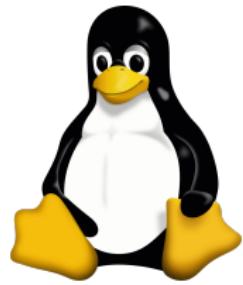
*Brasília-DF, 2025*

# Sumário

- Preliminares
- História do Linux
- GNU/Linux
- Distribuições
- Terminal e Comandos

# Referências

- Negus C. Linux Bible 10ed 2020
- Este material está licenciado conforme a Licença GPL
- Responsável: IEEE Computer Society UnB



## Preliminares

# O que é um Sistema Operacional?

- É um software que atua como intermediário entre o usuário e o hardware do computador.

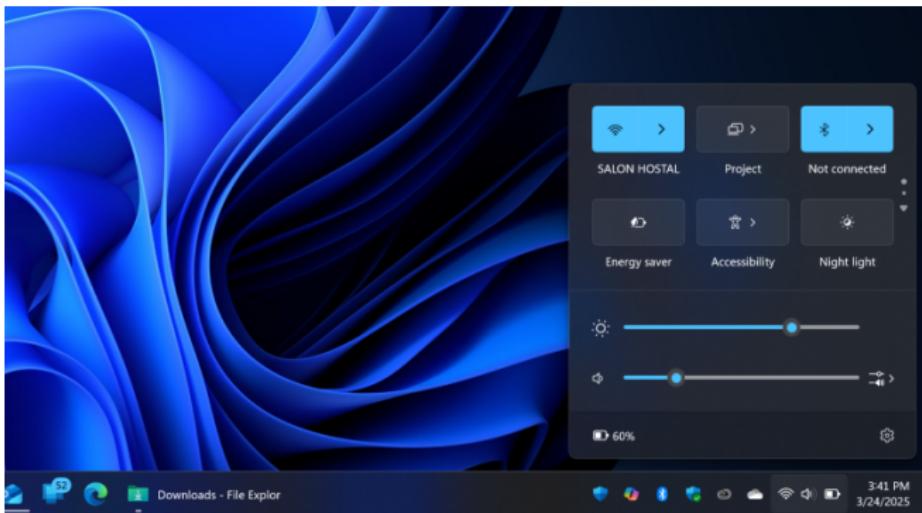
# O que é um Sistema Operacional?

- É um software que atua como intermediário entre o usuário e o hardware do computador.
- Responsável por gerenciar recursos como CPU, memória, dispositivos de entrada/saída e arquivos.

# O que é um Sistema Operacional?

- É um software que atua como intermediário entre o usuário e o hardware do computador.
- Responsável por gerenciar recursos como CPU, memória, dispositivos de entrada/saída e arquivos.
- Permite que os programas sejam executados de forma eficiente e segura.

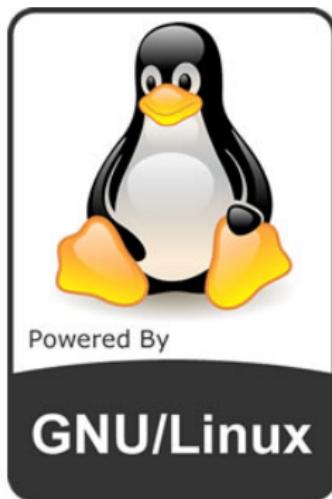
É fácil lembrar do...

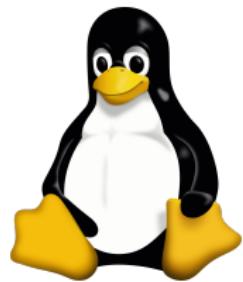


# Exemplos



Opa...





## História do GNU/Linux

# O que é o Unix?

- Unix é um sistema operacional criado em 1969 por Ken Thompson, Dennis Ritchie e outros desenvolvedores na AT&T Bell Labs.



# O que é o Unix?

- Foi projetado para ser um sistema multitarefa e multiusuário.

# O que é o Unix?

- Unix é um sistema operacional criado em 1969 por Ken Thompson, Dennis Ritchie e outros desenvolvedores na AT&T Bell Labs.
- Foi projetado para ser um sistema multitarefa e multiusuário.

# O que é o Unix?

- Unix é um sistema operacional criado em 1969 por Ken Thompson, Dennis Ritchie e outros desenvolvedores na AT&T Bell Labs.
- Foi projetado para ser um sistema multitarefa e multiusuário.
- Escrito originalmente em linguagem de montagem, depois foi reescrito em C, facilitando sua portabilidade.

# O que é o Unix?

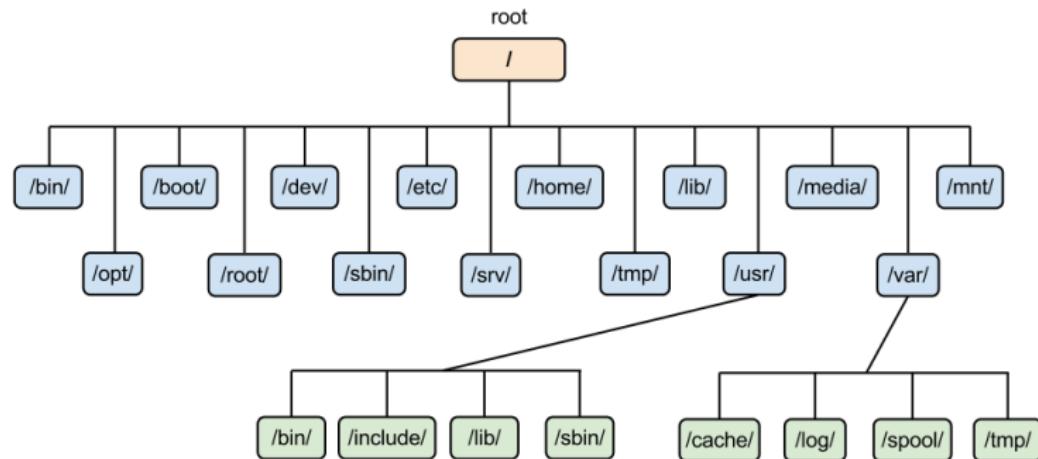


# A Filosofia Unix

- Filosofia do Unix: programas pequenos, simples e modulares.

# A Filosofia Unix

- Filosofia do Unix: programas pequenos, simples e modulares.
- Utiliza uma estrutura de arquivos hierárquica e...



# A Filosofia Unix

- Filosofia do Unix: programas pequenos, simples e modulares.
- Utiliza uma estrutura de arquivos hierárquica e possui comandos em modo texto muito poderosos.

```
$PP 13:53 .
. Sep 15:53 ..
0. Sep 2015 bin -> usr/bin
19. Sep 09:31 boot
21. Sep 15:58 dev
19. Sep 09:32 etc
21. Sep 15:52 home
7 30. Sep 2015 lib -> usr/lib
7 30. Sep 2015 lib64 -> usr/lib
84 23. Jul 10:01 lost+found
96 1. Aug 22:45 mnt
996 38. Sep 2015 opt
6 21. Sep 15:52 private -> /home/encrypted
4096 12. Aug 15:37 proc
568 21. Sep 15:58 root
7 30. Sep 15:58 run
4096 30. Sep 2015 sbin -> usr/bin
6 21. Sep 2015 srv -> usr/bin
300 21. Sep 15:51 sys
4096 12. Aug 15:45 tmp
4096 23. Jul 10:25 usr
1000 4096 21. Sep 15:52 var
1000 4096 21. Sep 15:52
```

# A Filosofia Unix

- Filosofia do Unix: programas pequenos, simples e modulares.
- Utiliza uma estrutura de arquivos hierárquica e possui comandos em modo texto muito poderosos.
- Muitos conceitos do Unix são usados até hoje em servidores, redes e segurança da informação.

# Do Unix ao Linux

- O Unix estabeleceu os conceitos fundamentais dos sistemas operacionais modernos.

# Do Unix ao Linux

- O Unix estabeleceu os conceitos fundamentais dos sistemas operacionais modernos.
- Mas... nos anos 80 e 90, versões completas do Unix eram proprietárias e caras.



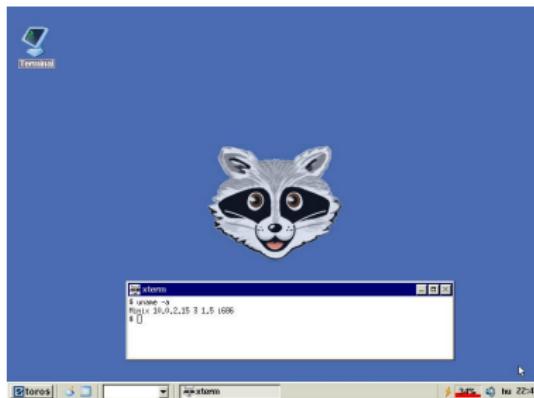
# Do Unix ao Linux

- Linus Torvalds, estudante na Finlândia, queria um sistema similar ao Unix, mas gratuito.



# Do Unix ao Linux

- Linus Torvalds, estudante na Finlândia, queria um sistema similar ao Unix, mas gratuito.
- Em 1991, ele começou a desenvolver seu próprio kernel no **Minix**, um sistema educacional baseado no Unix.



# O que é o Kernel?



# O que é o Kernel?

- O kernel é o núcleo central de um sistema operacional.

# O que é o Kernel?

- O kernel é o núcleo central de um sistema operacional.
- Ele faz a ponte entre o hardware e os softwares em execução.

# O que é o Kernel?

- O kernel é o núcleo central de um sistema operacional.
- Ele faz a ponte entre o hardware e os softwares em execução.
- Gerencia recursos do sistema como CPU, memória, dispositivos e chamadas de sistema.

# O que é o Kernel?

- O kernel é o núcleo central de um sistema operacional.
- Ele faz a ponte entre o hardware e os softwares em execução.
- Gerencia recursos do sistema como CPU, memória, dispositivos e chamadas de sistema.
- No Linux, o kernel é modular e pode ser personalizado ou recompilado.

# Ops...

- Voltando pro Linux...

# Do Unix ao Linux

- Linus Torvalds, estudante na Finlândia, queria um sistema similar ao Unix, mas gratuito.
- Em 1991, ele começou a desenvolver seu próprio kernel no **Minix**, um sistema educacional baseado no Unix.
- Torvalds lançou o Linux como um projeto **open source**, permitindo que a comunidade contribuísse.

# Open Source

- O código-fonte está disponível publicamente para leitura, modificação e redistribuição.

# Open Source

- O código-fonte está disponível publicamente para leitura, modificação e redistribuição.
- Enfatiza os benefícios técnicos da colaboração aberta.

# Open Source

- O código-fonte está disponível publicamente para leitura, modificação e redistribuição.
- Enfatiza os benefícios técnicos da colaboração aberta.
- Nem todo open source é software livre (diferenças filosóficas e legais).

# Open Source

- O código-fonte está disponível publicamente para leitura, modificação e redistribuição.
- Enfatiza os benefícios técnicos da colaboração aberta.
- Nem todo open source é **software livre** (diferenças filosóficas e legais).



# Software Livre

- Software que respeita a liberdade dos usuários.

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:
  - Executar o programa para qualquer propósito.

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:
  - Executar o programa para qualquer propósito.
  - Estudar como o programa funciona e modificá-lo.

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:
  - Executar o programa para qualquer propósito.
  - Estudar como o programa funciona e modificá-lo.
  - Redistribuir cópias.

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:
  - Executar o programa para qualquer propósito.
  - Estudar como o programa funciona e modificá-lo.
  - Redistribuir cópias.
  - Melhorar o programa e liberar suas melhorias.

# Software Livre

- Software que respeita a liberdade dos usuários.
- Usuários podem:
  - Executar o programa para qualquer propósito.
  - Estudar como o programa funciona e modificá-lo.
  - Redistribuir cópias.
  - Melhorar o programa e liberar suas melhorias.
- **Não é sinônimo de software gratuito.**

# Free Software



# Software Proprietário

- Software cujo código-fonte não é acessível ao público.

# Software Proprietário

- Software cujo código-fonte não é acessível ao público.
- Os usuários têm licenças restritas: podem usar, mas não modificar ou redistribuir.

# Software Proprietário

- Software cujo código-fonte não é acessível ao público.
- Os usuários têm licenças restritas: podem usar, mas não modificar ou redistribuir.
- Comum em empresas que protegem sua propriedade intelectual.

# Software Proprietário



# Do Unix ao Linux

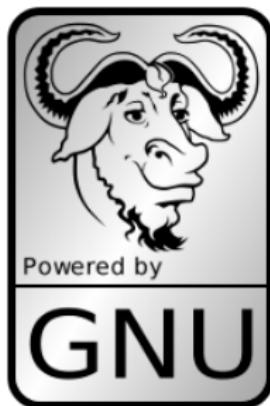
- O Unix estabeleceu os conceitos fundamentais dos sistemas operacionais modernos.
- Inspirado pelo Unix, o Linux surgiu como uma alternativa de código aberto e gratuita.
- Linus Torvalds criou o kernel do Linux em 1991, com base nos princípios do Unix.
- Linux mantém compatibilidade com muitos comandos e estruturas do Unix.

# O surgimento do Linux

- O Linux foi originalmente lançado sob uma licença restrita.

# O surgimento do Linux

- O Linux foi originalmente lançado sob uma licença restrita.
- Mas em 1992, Linux foi relicenciado sob a GNU GPL (General Public License).



# A História do Projeto GNU

- Em 1983, **Richard Stallman** lançou o **Projeto GNU** (GNU's Not Unix).



# A História do Projeto GNU

- Em 1983, **Richard Stallman** lançou o **Projeto GNU** (GNU's Not Unix).
- Seu objetivo era criar um sistema operacional totalmente livre, inspirado no Unix.



# A História do Projeto GNU

- Em 1983, **Richard Stallman** lançou o **Projeto GNU** (GNU's Not Unix).
- Seu objetivo era criar um sistema operacional totalmente livre, inspirado no Unix.
- Em 1985, Stallman fundou a **Free Software Foundation (FSF)** para apoiar o movimento.



# Stallman na UnB



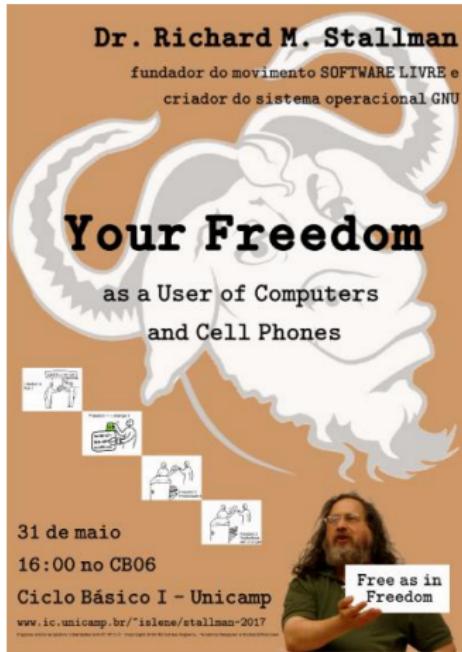
[https://gesporte.blogspot.com/2012/12/  
richard-stallman-realiza-palestra-na.html](https://gesporte.blogspot.com/2012/12/richard-stallman-realiza-palestra-na.html)

# Stallman na UnB



<https://www.youtube.com/watch?v=tZhqlpNNK-g>

# Stallman no Brasil



[https://unicamp.br/unicamp/noticias/2017/05/25/  
richard-stallman-fundador-do-movimento-software-livre-faz-palestra-na-](https://unicamp.br/unicamp/noticias/2017/05/25/richard-stallman-fundador-do-movimento-software-livre-faz-palestra-na-)

# Stallman no Brasil



[https://unicamp.br/unicamp/noticias/2017/05/25/  
richard-stallman-fundador-do-movimento-software-livre-faz-palestra-na-](https://unicamp.br/unicamp/noticias/2017/05/25/richard-stallman-fundador-do-movimento-software-livre-faz-palestra-na-)

# A História do Projeto GNU

- O Projeto GNU desenvolveu ferramentas essenciais:

# A História do Projeto GNU

- O Projeto GNU desenvolveu ferramentas essenciais:
  - Compilador GCC;

# A História do Projeto GNU

- O Projeto GNU desenvolveu ferramentas essenciais:
  - Compilador `GCC`;
  - Editor `Emacs`;

# A História do Projeto GNU

- O Projeto GNU desenvolveu ferramentas essenciais:
  - Compilador **GCC**;
  - Editor **Emacs**;
  - shell **Bash**, entre outros.

# A História do Projeto GNU

- O Projeto GNU desenvolveu ferramentas essenciais:
  - Compilador `GCC`;
  - Editor `Emacs`;
  - shell `Bash`, entre outros.
- Contudo, o sistema GNU não tinha um **kernel funcional**.

# Faltava um Kernel Livre

- O Projeto GNU tinha quase todos os componentes de um sistema operacional...

# Faltava um Kernel Livre

- O Projeto GNU tinha quase todos os componentes de um sistema operacional...
- ... mas faltava o kernel (núcleo do sistema).

# Faltava um Kernel Livre

- O Projeto GNU tinha quase todos os componentes de um sistema operacional...
- ... mas faltava o kernel (núcleo do sistema).
- O kernel Hurd estava em desenvolvimento, mas era instável e lento.

# GNU + Linux = GNU/Linux

- Com o Linux licenciado como livre, tornou-se possível combiná-lo com o restante do sistema GNU.



# GNU + Linux = GNU/Linux

- Com o Linux licenciado como livre, tornou-se possível combiná-lo com o restante do sistema GNU.
- Surgiu, assim, o sistema operacional completo conhecido como **GNU/Linux**.



GNU + Linux = GNU/Linux

**GNU**



**LINUX**



**GNU  
+  
LINUX**



gnu meets linux

# GNU + Linux = GNU/Linux

- O nome (**GNU/Linux**) reconhece a contribuição do Projeto GNU.
- Stallman defende o uso do termo para dar crédito ao movimento Software Livre.
- Ainda há debate na comunidade: muitos simplesmente chamam de “Linux”.

# GNU + Linux = GNU/Linux



# Impacto no mundo

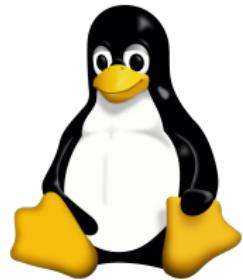
- GNU/Linux domina servidores, dispositivos embarcados e smartphones (Android).

# Impacto no mundo

- GNU/Linux domina servidores, dispositivos embarcados e smartphones (Android).
- É a base de projetos de soberania tecnológica e educação.

# Impacto no mundo

- GNU/Linux domina servidores, dispositivos embarcados e smartphones (Android).
- É a base de projetos de soberania tecnológica e educação.
- Incentiva valores como liberdade, transparência e colaboração.



# GNU/Linux

# Modo Texto e Modo Gráfico

- O Linux pode ser usado de duas formas principais:

# Modo Texto e Modo Gráfico

- O Linux pode ser usado de duas formas principais:
  - Modo texto (console ou terminal)**

```
Ubuntu 18.04.2 LTS ubuntuserver tty1

ubuntuserver login: sk
Password:
Last login: Thu Aug  8 17:32:34 IST 2019 on tty1
[ 42.211710] cloud-init[1271]: Cloud-init v. 19.1-1-gbaa47854-0ubuntu1~18.04.1 running 'modules:config' at Thu, 08 Aug 2019 12:08:44 +0000. Up 41.83 seconds.
[ 44.230735] cloud-init[1359]: Cloud-init v. 19.1-1-gbaa47854-0ubuntu1~18.04.1 running 'modules:final' at Thu, 08 Aug 2019 12:08:46 +0000. Up 43.55 seconds.
[ 44.231178] cloud-init[1359]: Cloud-init v. 19.1-1-gbaa47854-0ubuntu1~18.04.1 finished at Thu, 08 Aug 2019 12:08:47 +0000. Datasource DataSourceNoCloud [seed=/var/lib/cloud/seed/nocloud-net] [dsmod=net]. Up 44.19 seconds
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

* Keen to learn Istio? It's included in the single-package MicroK8s.

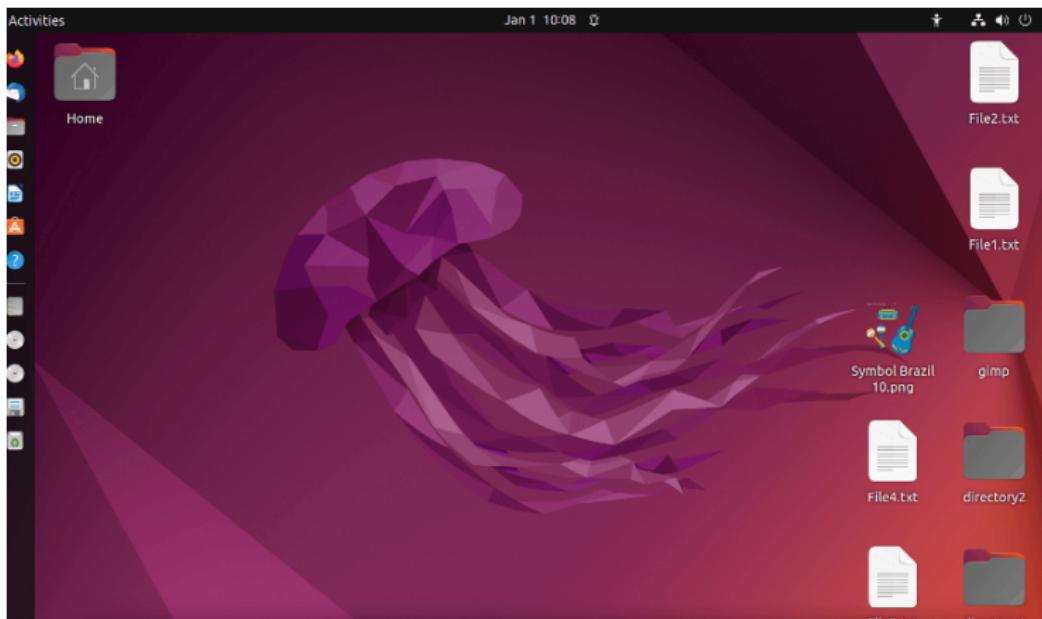
      https://snapcraft.io/microk8s

44 packages can be updated.
0 updates are security updates.

sk@ubuntuserver:~$
```

# Modo Texto e Modo Gráfico

- O Linux pode ser usado de duas formas principais:
  - ➊ Modo gráfico (interface de desktop)



# Modo Texto e Modo Gráfico

- O Linux pode ser usado de duas formas principais:
  - ① **Modo texto (console ou terminal)**
  - ② **Modo gráfico (interface de desktop)**
- Ambas coexistem no sistema e podem ser usadas conforme a necessidade.

# Modo Texto (Modo Console)

- Baseada em linha de comando (CLI - *Command Line Interface*).

# Modo Texto (Modo Console)

- Baseada em linha de comando (CLI - *Command Line Interface*).
- Utiliza interpretadores de comandos chamados **shells** (ex: **bash**, **zsh**).

# Modo Texto (Modo Console)

- Baseada em linha de comando (CLI - *Command Line Interface*).
- Utiliza interpretadores de comandos chamados **shells** (ex: **bash**, **zsh**).
- Vantagens:
  - Leve e rápida
  - Ideal para servidores e automação
  - Controle preciso do sistema

# Modo Texto (Modo Console)

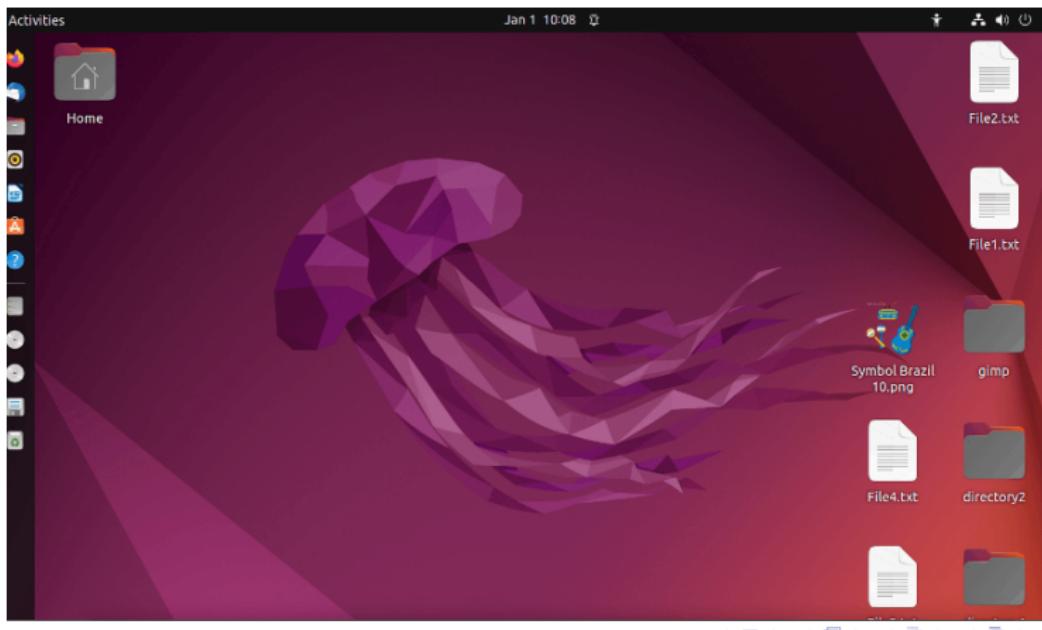
- Baseada em linha de comando (CLI - *Command Line Interface*).
- Utiliza interpretadores de comandos chamados **shells** (ex: **bash**, **zsh**).
- Vantagens:
  - Leve e rápida
  - Ideal para servidores e automação
  - Controle preciso do sistema
- Ferramentas (comandos) comuns: **ls**, **cd**, **nano**, **apt**, **grep**, etc.

# Modo Gráfico (Interface Gráfica)

- Permite interação com janelas, ícones e menus.

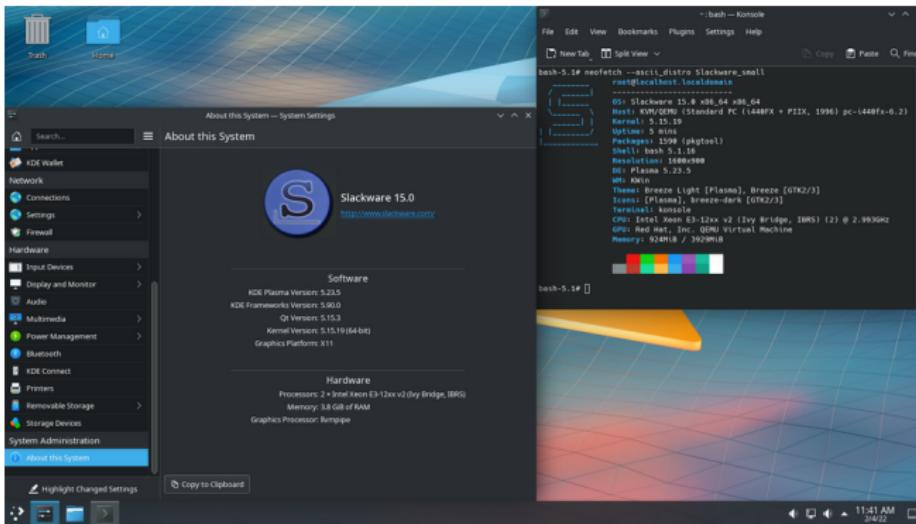
# Modo Gráfico (Interface Gráfica)

- Permite interação com janelas, ícones e menus.
- Tipos de ambientes gráficos (Desktops)
  - **GNOME**: Interface moderna e limpa (usada no Ubuntu padrão)



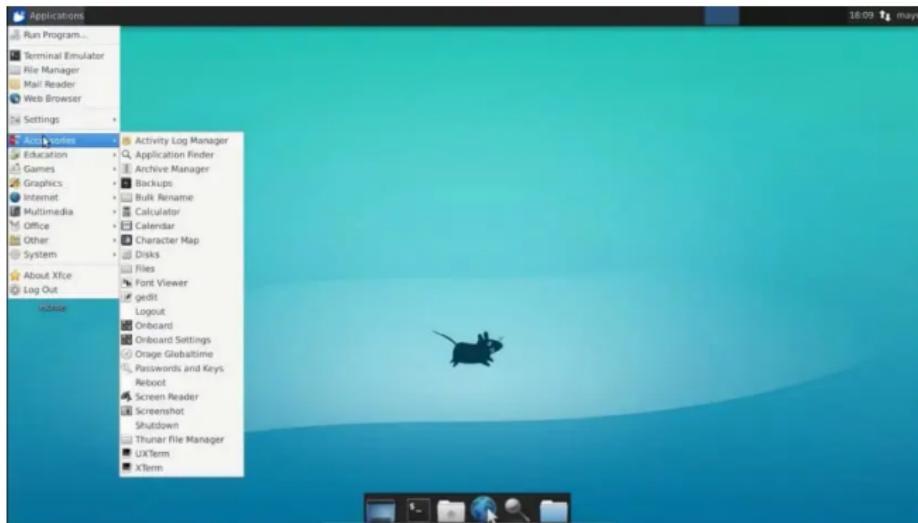
# Modo Gráfico (Interface Gráfica)

- Permite interação com janelas, ícones e menus.
- Tipos de ambientes gráficos (Desktops)
  - **KDE Plasma:** Altamente customizável e visualmente avançado



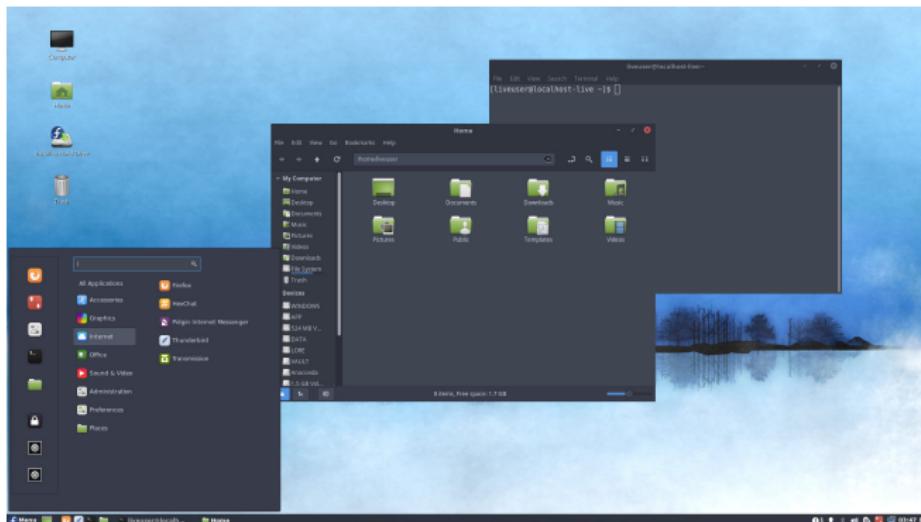
# Modo Gráfico (Interface Gráfica)

- Permite interação com janelas, ícones e menus.
- Tipos de ambientes gráficos (Desktops)
  - **Xfce**: Leve, ideal para computadores modestos



# Modo Gráfico (Interface Gráfica)

- Permite interação com janelas, ícones e menus.
- Tipos de ambientes gráficos (Desktops)
  - **Cinnamon**: Foco em familiaridade (usado no Linux Mint)



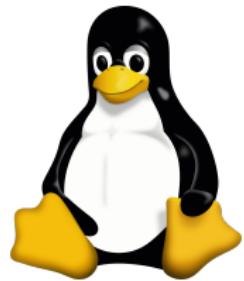
# Modo texto vs modo gráfico

- **Modo Texto:**

- Administração de servidores
- Scripts e automação
- Diagnóstico de problemas

- **Modo Gráfico:**

- Usuário comum
- Tarefas multimídia e produtividade
- Navegação, edição de documentos, etc.



## Distribuições Linux

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**
  - Ferramentas GNU (como Bash, coreutils)

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**
  - Ferramentas GNU (como Bash, coreutils)
  - Gerenciador de pacotes

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**
  - Ferramentas GNU (como Bash, coreutils)
  - Gerenciador de pacotes
  - Interface gráfica (opcional)

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**
  - Ferramentas GNU (como Bash, coreutils)
  - Gerenciador de pacotes
  - Interface gráfica (opcional)
  - Conjunto de softwares padrão

# O que são Distribuições Linux?

- Distribuições (**distros**) são versões completas do sistema operacional GNU/Linux.
- Elas incluem:
  - O **kernel Linux**
  - Ferramentas GNU (como Bash, coreutils)
  - Gerenciador de pacotes
  - Interface gráfica (opcional)
  - Conjunto de softwares padrão

# Cardápio de Distribuições



# O que diferencia as distribuições?

- Gerenciador de pacotes (ex: apt, dnf, installpkg)

# O que diferencia as distribuições?

- **Gerenciador de pacotes** (ex: apt, dnf, installpkg)
- **Foco do público-alvo** (usuários finais, servidores, segurança, etc.)

# O que diferencia as distribuições?

- **Gerenciador de pacotes** (ex: apt, dnf, installpkg)
- **Foco do público-alvo** (usuários finais, servidores, segurança, etc.)
- **Ambiente gráfico padrão** (GNOME, KDE, XFCE, etc.)

# O que diferencia as distribuições?

- **Gerenciador de pacotes** (ex: apt, dnf, installpkg)
- **Foco do público-alvo** (usuários finais, servidores, segurança, etc.)
- **Ambiente gráfico padrão** (GNOME, KDE, XFCE, etc.)
- **Frequência de atualização** (rolling release vs versões estáveis)

# O que diferencia as distribuições?

- **Gerenciador de pacotes** (ex: apt, dnf, installpkg)
- **Foco do público-alvo** (usuários finais, servidores, segurança, etc.)
- **Ambiente gráfico padrão** (GNOME, KDE, XFCE, etc.)
- **Frequência de atualização** (rolling release vs versões estáveis)
- **Licenciamento e filosofia** (ex: 100% livre vs inclusão de drivers proprietários)

# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.

# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.



# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.
- Baseado no Debian, com foco em:



# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.
- Baseado no Debian, com foco em:
  - Facilidade de uso

# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.
- Baseado no Debian, com foco em:
  - Facilidade de uso
  - Acessibilidade para usuários iniciantes

# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.
- Baseado no Debian, com foco em:
  - Facilidade de uso
  - Acessibilidade para usuários iniciantes
  - Suporte corporativo e comunitário

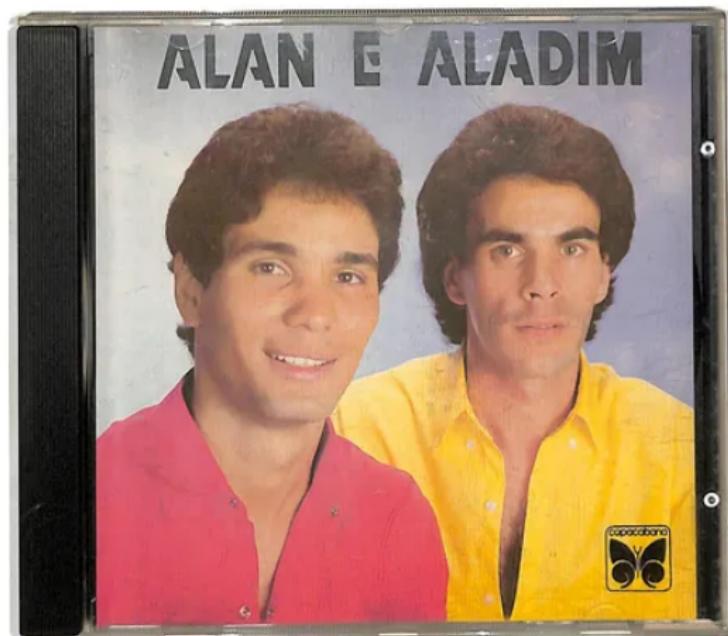
# O que é o Ubuntu?

- Ubuntu é uma das distribuições GNU/Linux mais populares do mundo.
- Criado em 2004 pela empresa **Canonical Ltd.**, fundada por **Mark Shuttleworth**.
- Baseado no Debian, com foco em:
  - Facilidade de uso
  - Acessibilidade para usuários iniciantes
  - Suporte corporativo e comunitário
- O nome “Ubuntu” é derivado de um conceito africano que significa “humanidade para os outros”.

# Antigamente...



# Último CD do Ubuntu



# Características principais

- Interface gráfica padrão: **GNOME** (anteriormente Unity).

# Características principais

- Interface gráfica padrão: **GNOME** (anteriormente Unity).
- Ciclo de lançamentos:
  - Nova versão a cada 6 meses
  - Versões LTS (Long Term Support) a cada 2 anos com suporte por 5 anos

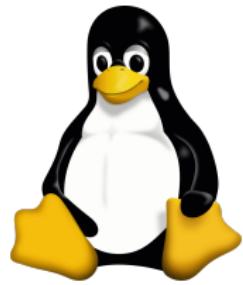
# Características principais

- Interface gráfica padrão: **GNOME** (anteriormente Unity).
- Ciclo de lançamentos:
  - Nova versão a cada 6 meses
  - Versões LTS (Long Term Support) a cada 2 anos com suporte por 5 anos
- Gerenciador de pacotes: APT, baseado em arquivos .deb

# Características principais

- Compatibilidade com “softwares” e *drivers* proprietários.

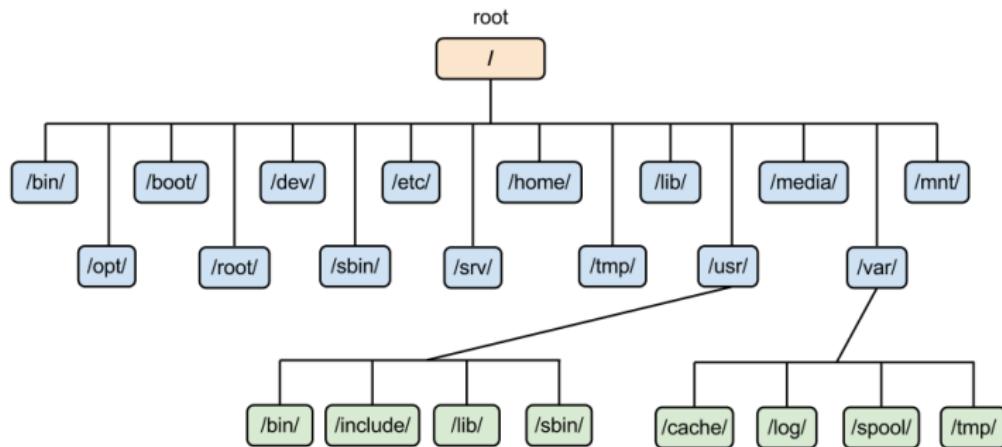




## Terminal e Comandos

# Sistema de Arquivos no Linux/Ubuntu

- O Linux usa um sistema de arquivos hierárquico com raiz em / (barra).



# Sistema de Arquivos no Linux/Ubuntu

- O Linux usa um sistema de arquivos hierárquico com raiz em / (barra).
- Todos os arquivos e diretórios fazem parte de uma única árvore.

# Sistema de Arquivos no Linux/Ubuntu

- O Linux usa um sistema de arquivos hierárquico com raiz em / (barra).
- Todos os arquivos e diretórios fazem parte de uma única árvore.

# Sistema de Arquivos no Linux/Ubuntu

- O Linux usa um sistema de arquivos hierárquico com raiz em / (barra).
- Todos os arquivos e diretórios fazem parte de uma única árvore.
- O sistema de arquivos mais comum no Ubuntu é o **ext4**.

# Montagem de Dispositivos no Linux

- No Linux, dispositivos como HDs, SSDs, pendrives e DVDs não aparecem automaticamente com letras (como no Windows).
- Eles são integrados ao sistema de arquivos por meio de um processo chamado **montagem** (**mount**).
- Montar significa **ligar o conteúdo de um dispositivo a um diretório existente**.
- Exemplo:

```
sudo mount /dev/sdb1 /mnt/usb
```

# Montagem de Dispositivos no Linux

- No Linux, dispositivos como HDs, SSDs, pendrives e DVDs não aparecem automaticamente com letras (como no Windows).

# Montagem de Dispositivos no Linux

- No Linux, dispositivos como HDs, SSDs, pendrives e DVDs não aparecem automaticamente com letras (como no Windows).
- Eles são integrados ao sistema de arquivos por meio de um processo chamado **montagem** (`mount`).

# Montagem de Dispositivos no Linux

- No Linux, dispositivos como HDs, SSDs, pendrives e DVDs não aparecem automaticamente com letras (como no Windows).
- Eles são integrados ao sistema de arquivos por meio de um processo chamado **montagem** (**mount**).
- Montar significa **ligar o conteúdo de um dispositivo a um diretório existente**.
- Exemplo:

```
sudo mount /dev/sdb1 /mnt/usb
```

# Montagem de Dispositivos no Linux

- Após a montagem, os arquivos podem ser acessados normalmente em `/mnt/usb`.
- Para remover com segurança, usa-se `umount`:

```
sudo umount /mnt/usb
```

# Diretórios principais

- /: Diretório raiz

# Diretórios principais

- `/`: Diretório raiz
- `/home`: Diretórios dos usuários

# Diretórios principais

- **/**: Diretório raiz
- **/home**: Diretórios dos usuários
- **/etc**: Arquivos de configuração do sistema

# Diretórios principais

- `/`: Diretório raiz
- `/home`: Diretórios dos usuários
- `/etc`: Arquivos de configuração do sistema
- `/bin`, `/usr/bin`: Executáveis essenciais

# Diretórios principais

- **/**: Diretório raiz
- **/home**: Diretórios dos usuários
- **/etc**: Arquivos de configuração do sistema
- **/bin, /usr/bin**: Executáveis essenciais
- **/var**: Arquivos variáveis (logs, spool)

# Diretórios principais

- **/**: Diretório raiz
- **/home**: Diretórios dos usuários
- **/etc**: Arquivos de configuração do sistema
- **/bin, /usr/bin**: Executáveis essenciais
- **/var**: Arquivos variáveis (logs, spool)
- **/tmp**: Arquivos temporários

# Diretórios principais

- `/`: Diretório raiz
- `/home`: Diretórios dos usuários
- `/etc`: Arquivos de configuração do sistema
- `/bin`, `/usr/bin`: Executáveis essenciais
- `/var`: Arquivos variáveis (logs, spool)
- `/tmp`: Arquivos temporários
- `/dev`: Dispositivos como arquivos (ex: `/dev/sda`)

# Diretórios principais

- `/`: Diretório raiz
- `/home`: Diretórios dos usuários
- `/etc`: Arquivos de configuração do sistema
- `/bin`, `/usr/bin`: Executáveis essenciais
- `/var`: Arquivos variáveis (logs, spool)
- `/tmp`: Arquivos temporários
- `/dev`: Dispositivos como arquivos (ex: `/dev/sda`)
- `/mnt`: `/media`: Pontos de montagem de dispositivos externos

# Usuários no Linux

- Todo acesso no Linux é feito por meio de um **usuário**.

# Usuários no Linux

- Todo acesso no Linux é feito por meio de um **usuário**.
- Existem dois tipos principais:
  - **Usuários comuns**: possuem permissões limitadas.
  - **Usuário root**: superusuário com controle total do sistema.

# Usuários no Linux

- Cada usuário possui:
  - Um diretório pessoal (ex: `/home/aluno`)
  - Um identificador numérico (UID)
  - Um grupo padrão
- As permissões de arquivos e comandos variam de acordo com o usuário.

# Comandos sudo

- `sudo` permite que um usuário autorizado execute comandos como root.
- Exemplo:

```
sudo adduser viniciusborges
```

# Comandos sudo

- `sudo` permite que um usuário autorizado execute comandos como root.
- Exemplo:

```
sudo adduser viniciusborges
```

- Atenção: nunca utilize o root no cotidiano.

# Comandos sudo

- **sudo** permite que um usuário autorizado execute comandos como root.
- Exemplo:

```
sudo adduser viniciusborges
```

- Atenção: nunca utilize o root no cotidiano.
- Prefira **sudo** para maior controle e segurança.

# Comando ls

- Lista arquivos e diretórios no diretório atual.
- Para listar os arquivos no diretório atual:

```
ls
```

- Para listar os arquivos no diretório “Documents”:

```
ls Documents
```

# Comando ls

- O comando:

```
ls -l
```

- Mostra permissões, dono, tamanho e data de modificação.
- Use ls -a para ver arquivos ocultos.

# Comando cd

- Navega entre diretórios.
- Exemplos:

```
cd /home/aluno
```

```
cd ..          (volta um diretório)
```

- Use cd para ir para o diretório do usuário.

# Comando pwd

- Mostra o caminho completo (absoluto) do diretório atual.
- Exemplo:

```
pwd
```

- pode retornar algo como:

```
/home/aluno/projetos
```

- Útil para saber onde você está no sistema de arquivos.

## Comando df -h

- Exibe o uso do disco (espaço ocupado/livre).
- Exemplo:

```
df -h
```

- A opção -h mostra os valores em GB/MB.

## Comando du -sh

- Mostra o tamanho de um diretório específico.
- Exemplo:

```
du -sh /Downloads
```

- Útil para identificar pastas que ocupam muito espaço.

## Comando find

- Localiza arquivos e diretórios de forma recursiva.
- Exemplo:

```
find /home -name arquivo.txt
```

- Suporta filtros por tipo, data e tamanho.

# Comando locate

- Busca rápida de arquivos usando um banco de dados indexado.
- Exemplo:

```
locate config.json
```

- Atualize o índice com `sudo updatedb`.

# Comando `mkdir`

- Cria diretórios (pastas) no sistema de arquivos.
- Exemplo:

```
mkdir projetos
```

# Comando `mkdir`

- Para criar diretórios aninhados:

```
mkdir -p curso/linux/aulas
```

- O argumento `-p` cria todos os diretórios intermediários.

# Comando rm

- Remove arquivos ou diretórios.
- Exemplo para arquivos:

```
rm arquivo.txt
```

- Para remover diretórios e seu conteúdo:

```
rm -r pasta/
```

## Comando `rm`

- Use com `-f` para forçar a remoção sem confirmação: `rm -rf pasta/`
- **CUIDADO:** não há lixeira. Executou `rm`, já era!

# Comando mv

- Move arquivos e diretórios ou altera seus nomes.
- Para mover:

```
mv relatorio.txt /home/aluno/documentos/
```

- Para renomear:

```
mv antigo.txt novo.txt
```

- Substitui o arquivo de destino, se já existir.

# Comando cp

- Copia arquivos ou diretórios.
- Exemplo simples:

```
cp arquivo.txt copia.txt
```

- Para copiar diretórios com conteúdo:

```
cp -r pasta/ backup/
```

- O argumento **-r** (recursivo) é necessário para copiar diretórios.

# Comando cat

- **cat** vem de “concatenate” (concatenar).
- É usado para:
  - Exibir o conteúdo de arquivos no terminal
  - Concatenar arquivos em sequência
  - Criar arquivos simples a partir da entrada do usuário
- É um dos comandos mais usados em scripts e inspeção rápida de arquivos.

# Comando cat

- **cat arquivo.txt**  
Exibe o conteúdo de arquivo.txt
- **cat arquivo1.txt arquivo2.txt > combinado.txt**  
Junta dois arquivos em um novo arquivo
- **cat > novo.txt**  
Cria um arquivo novo a partir da entrada no terminal  
(finalize com Ctrl+D)

# Comando cat

- `cat arquivo.txt | less`  
Exibe o conteúdo com rolagem interativa
- `cat arquivo.txt | grep "erro"`  
Filtrar linhas que contenham “erro”

# Editor de texto nano

- **nano** é um editor de texto simples e fácil de usar no terminal.

# Editor de texto nano

- **nano** é um editor de texto simples e fácil de usar no terminal.
- Ideal para iniciantes que precisam editar arquivos de forma rápida e direta.

# Editor de texto nano

- **nano** é um editor de texto simples e fácil de usar no terminal.
- Ideal para iniciantes que precisam editar arquivos de forma rápida e direta.
- Exibe os comandos mais usados na parte inferior da tela.

# Editor de texto nano

- **nano** é um editor de texto simples e fácil de usar no terminal.
- Ideal para iniciantes que precisam editar arquivos de forma rápida e direta.
- Exibe os comandos mais usados na parte inferior da tela.

# Editor de texto nano

- Exemplos de uso:

```
nano arquivo.txt
```

- Para salvar: Ctrl + O
- Para sair: Ctrl + X

# Editor de texto nano

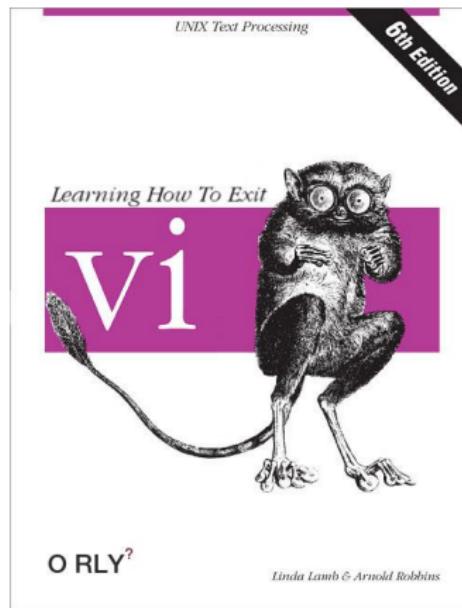
- Exemplos de uso:

```
nano arquivo.txt
```

- Para salvar: Ctrl + O
- Para sair: Ctrl + X

# E o Editor vi?

- Espere o curso avançado de GNU/Linux da IEEE Computer Society UnB!



# Comando grep

- O comando **grep** busca padrões de texto em arquivos ou na saída de outros comandos.
- Nome vem de: **Global Regular Expression Print**.
- É uma ferramenta poderosa para filtragem de conteúdo com base em expressões regulares.
- Usos comuns:
  - Encontrar palavras específicas em arquivos
  - Filtrar linhas em comandos como **cat**, **ps**, **dmesg**
  - Buscar por padrões (ex: datas, IPs, erros)

# Comando grep

- `grep "erro" log.txt`  
Mostra linhas com a palavra “erro” no arquivo
- `cat arquivo.txt | grep "palavra"`  
Filtrar linhas com “palavra” na saída do `cat`

# Comando grep

- `grep -i "usuario" dados.txt`  
Busca ignorando maiúsculas/minúsculas
- `grep -r "main" .`  
Busca recursivamente por “main” em todos os arquivos
- `grep -n "import" script.py`  
Mostra o número das linhas onde o termo aparece

# Comando apt

- O comando `apt` é o gerenciador de pacotes usado no Ubuntu.

# Comando apt

- O comando **apt** é o gerenciador de pacotes usado no Ubuntu.
- Permite instalar, atualizar, remover e buscar pacotes de software.

# Comando apt

- O comando **apt** é o gerenciador de pacotes usado no Ubuntu.
- Permite instalar, atualizar, remover e buscar pacotes de software.
- Interage com repositórios oficiais e externos.

# Comando apt

- O comando **apt** é o gerenciador de pacotes usado no Ubuntu.
- Permite instalar, atualizar, remover e buscar pacotes de software.
- Interage com repositórios oficiais e externos.
- Substitui comandos mais antigos como **apt-get** em uso interativo.

# Comando apt

- O comando **apt** é o gerenciador de pacotes usado no Ubuntu.
- Permite instalar, atualizar, remover e buscar pacotes de software.
- Interage com repositórios oficiais e externos.
- Substitui comandos mais antigos como **apt-get** em uso interativo.
- Requer permissões de superusuário (**sudo**).

# Comando apt

- `sudo apt update`: atualiza a lista de pacotes disponíveis

## Comando apt

- `sudo apt update`: atualiza a lista de pacotes disponíveis
- `sudo apt upgrade`: atualiza os pacotes instalados

## Comando apt

- `sudo apt update`: atualiza a lista de pacotes disponíveis
- `sudo apt upgrade`: atualiza os pacotes instalados
- `sudo apt install nome-do-pacote`: instala um pacote

## Comando apt

- `sudo apt update`: atualiza a lista de pacotes disponíveis
- `sudo apt upgrade`: atualiza os pacotes instalados
- `sudo apt install nome-do-pacote`: instala um pacote
- `sudo apt remove nome-do-pacote`: remove um pacote

## Comando apt

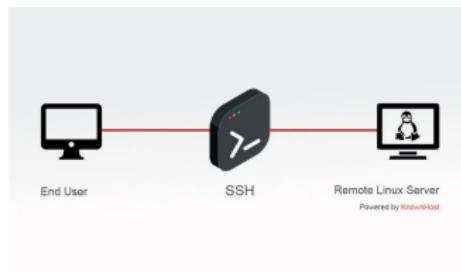
- `sudo apt update`: atualiza a lista de pacotes disponíveis
- `sudo apt upgrade`: atualiza os pacotes instalados
- `sudo apt install nome-do-pacote`: instala um pacote
- `sudo apt remove nome-do-pacote`: remove um pacote
- `apt search nome`: procura um pacote

# Comando apt

- `sudo apt update`: atualiza a lista de pacotes disponíveis
- `sudo apt upgrade`: atualiza os pacotes instalados
- `sudo apt install nome-do-pacote`: instala um pacote
- `sudo apt remove nome-do-pacote`: remove um pacote
- `apt search nome`: procura um pacote
- `apt show nome`: mostra informações detalhadas do pacote

# Comando ssh

- O comando **ssh** permite acessar outro computador de forma segura via rede.
- Utiliza criptografia para proteger a conexão.



# Comando ssh

- Sintaxe:

```
ssh usuario@ip-ou-dominio
```

- Exemplo:

```
ssh aluno@192.168.1.10
```

- Pode ser usado para acessar servidores, fazer manutenção, executar comandos remotos, etc.

# Comando scp

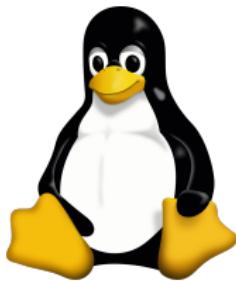
- O comando **scp** copia arquivos entre máquinas usando o protocolo SSH.
- Sintaxe básica:

```
scp origem destino
```

- Exemplos:
  - `scp arquivo.txt aluno@192.168.1.10:/home/aluno/`
  - `scp aluno@192.168.1.10:/var/log/syslog ./`
- Ideal para transferir arquivos de forma segura entre computadores remotos.

# Considerações Finais

- Atenção: este material abordou os comandos e conceitos básicos do GNU/Linux.
- Para aprofundar, explore tópicos intermediários e avançados:
  - Permissões avançadas e controle de grupos (`chmod`, `chown`, `umask`)
  - Redirecionamento e encadeamento de comandos (`>`, `>>`, `<`, `|`, `&`)
  - Scripts em `bash` (automatização de tarefas)
  - Gerenciamento de serviços com `systemd`
  - Agendamento de tarefas com `cron`
  - Gerenciamento de usuários e processos
  - Redes, firewall e segurança no Linux



## Minicurso de Introdução ao GNU/Linux

IEEE Computer Society UnB

*Brasília-DF, 2025*