

# CURSO DE INTRODUÇÃO AO LINUX

Distribuição openSUSE®

AULA 3 – Documentação; Operações de  
linha de comando

# Objetivos dessa aula

- Conhecer diferentes fontes de documentação;
- Utilizar as `man pages` (páginas de manual);
- Acessar o sistema `info` do GNU;
- Usar o comando `help` e a opção `-help`;
- Utilizar a linha de comando para realizar operações no Linux tais como:
  - Navegar entre diretórios;
  - Instalar e atualizar software.

# Documentação Linux

## *Introdução a Fontes de Documentação Linux*

Seja você um usuário inexperiente ou veterano, não saberá sempre como usar vários programas e ferramentas do Linux, ou o que digitar na linha de comando: você precisará consultar a documentação de ajuda regularmente.

Como sistemas Linux têm uma grande variedade de fontes, são numerosos as reservas de documentação e maneiras de conseguir ajuda. Os distribuidores consolidam esse material e o apresentam de uma maneira compreensível de usar.

Algumas fontes importantes de documentação são as seguintes:

- **man pages;**
- **GNU Info;**
- **Comando `help` e opção `--help`.**

# Documentação Linux

## *As páginas de manual (man pages)*

As páginas de manual são a mais comumente usada fonte de documentação no Linux. Elas providenciam documentação aprofundada sobre muitos programas e ferramentas, além de outros tópicos, incluindo arquivos de configuração, chamadas de sistema, rotinas de bibliotecas e o kernel.

Digitar *man* com o nome de tópico como argumento recupera a informação guardada nas páginas sobre este tópico. Algumas distribuições, inclusive, requerem que todo programa instalado tenha uma página man correspondente.

# Documentação Linux

## *man*

O programa *man* procura, formata e amostra a informação contida nas *man pages*. Como muitos tópicos têm muita informação, a saída é encaminhada através de uma janela do terminal, para ao menos uma página ser vista por vez. Ao mesmo tempo, as informações são formatadas para uma boa exibição visual.

Quando não são dadas opções, por padrão o usuário verá apenas a página dedicada especificamente sobre o tópico. É possível expandir a visualização para todas as páginas contendo a *string* em seu nome com a opção *-f*. Também é possível ver todas as páginas que discorrem sobre um tópico específico (mesmo que este não esteja no nome) com a opção *-k*.

# Documentação Linux

## *Capítulos dos manuais*

As man pages são divididas em nove capítulos, numerados de 1 a 9. Às vezes, uma letra é adicionada ao número do capítulo para identificar um tópico específico.

O número do capítulo pode ser usado para forçar o man a mostrar a página de um capítulo específico; é comum ter várias páginas entre vários capítulos com o mesmo nome, principalmente para nomes de funções de bibliotecas ou chamadas de sistema.

Com o parâmetro `-a`, man mostrará todas as páginas com um dado nome em todos os capítulos, um após o outro.

# Documentação Linux

## *Sistema GNU Info*

É o formato padrão de documentação do projeto GNU, que o prefere ao man. O sistema info tem forma mais livre e permite links entre subseções.

Funcionalmente, o info lembra o man em várias maneiras. Porém, os tópicos são conectados entre links (mesmo seu design precedendo a WWW).

Informações podem ser vistas tanto através de uma interface de linha de comando, uma ferramenta gráfica de ajuda, impressas ou online.

# Documentação Linux

## *Navegador Info pela CLI*

Digitar `info` sem argumentos em um terminal mostra um índice de tópicos disponíveis. Você pode navegar pela lista de tópicos usando as setas, Page Up e Page Down.

Você pode ver ajuda para um tópico particular digitando `info <nome>`. O sistema então procurará pelo tópico em todos os arquivos `info`.

Algumas teclas úteis são: `q` (sair), `h` (ajuda) e `Enter` (selecionar um item).



# Documentação Linux

## *Estrutura de uma página info*

O tópico que você vê na página info é chamado de **node** (nó).

Nós são similares a seções e subseções em documentações escritas. Você pode vasculhar entre nós ou ver cada nó sequencialmente. Cada nó pode conter menus e subtópicos linkados, ou itens.

Itens podem ser comparados a hyperlinks. Eles são identificados por um asterisco no começo do nome do item. Itens nomeados (fora de um menu) são identificados com um duplo dois-pontos (::) ao fim do nome do item.

Itens podem referenciar a outros nós no arquivo ou em outros arquivos. Alguns atalhos de navegação entre nós são os seguintes: **n** (próximo), **p** (anterior), **u** (subir um nó no índice).

# Documentação Linux

## *Opção --help*

A maioria dos comandos têm disponível uma breve descrição que pode ser vista usando a opção `--help` ou `-h` junto do comando ou aplicação. Por exemplo, você pode aprender mais sobre o comando `man` usando o seguinte comando:

- `$ man --help`

A opção `--help` é útil como uma referência simples e mostra informações mais rapidamente que as páginas `man` ou `info`.

# Documentação Linux

## *Comando help*

Alguns comandos populares (como `echo`), quando executados em um shell de comando `bash`, silenciosamente executam suas próprias versões integradas de programas ou ferramentas do sistema por ser mais eficiente fazê-lo. Para uma sinopse desses comandos integrados, você pode digitar `help`.

Para esses comandos integrados, `help` realizará a mesma função básica que os argumentos `-h` e `--help` realizam em aplicações independentes.

# Documentação Linux

## *Outras fontes*

Adicionalmente às páginas man, ao sistema GNU info e ao comando help, há outras fontes de documentação Linux, como:

- Sistema de ajuda na desktop (e.g. **khelppcenter**);
- Documentação de pacotes (encontrada no diretório `/usr/share/doc`);
- Recursos online, como:
  - LinuxCommand.org: <http://linuxcommand.org/tlcl.php>
  - OpenSUSE: <http://en.opensuse.org/Portal:Documentation>

# Operações em Linha de Comando

## *Introdução à CLI*

Administradores de sistemas Linux gastam um tempo considerável em um prompt de comando. Eles comumente automatizam tarefas e resolvem problemas nesse ambiente de texto.

Há um ditado: “interfaces gráficas facilitam tarefas fáceis, linhas de comando possibilitam tarefas difíceis”.

# Operações em Linha de Comando

## *Introdução à CLI*

O Linux depende severamente da abundância de ferramentas de linha de comando. A CLI possui as seguintes vantagens:

- Não há gasto de recursos com GUI;
- Virtualmente qualquer tarefa pode ser realizada pela linha de comando;
- Você pode automatizar tarefas e séries de procedimentos através de scripts;
- Você pode acessar remotamente máquinas em rede em qualquer lugar;
- Você pode iniciar aplicações gráficas diretamente da linha de comando.

# Operações em Linha de Comando

## *A Linha de comando*

A maior parte dos comandos inseridos em um prompt têm três elementos básicos:

- Comando
- Opções
- Argumentos

O *comando* é o nome do programa a ser executado. Pode ser seguido de uma ou mais *opções* (que começam com um ou dois traços, e.g. -h e --help) que modificam o que o comando pode fazer. Os *argumentos* representam aquilo em que o comando vai operar.

Vários comandos, porém, não possuem nem opções, nem argumentos.

# Operações em Linha de Comando

## *sudo*

O `sudo` permite aos usuários executarem programas com os privilégios de segurança de outro usuário, geralmente `root` (superusuário). A funcionalidade do `sudo` é tal qual *Executar como...* no Windows.

Em algumas distribuições, talvez seja necessário configurar e habilitar o `sudo` para funcionar corretamente.



# Operações em Linha de Comando

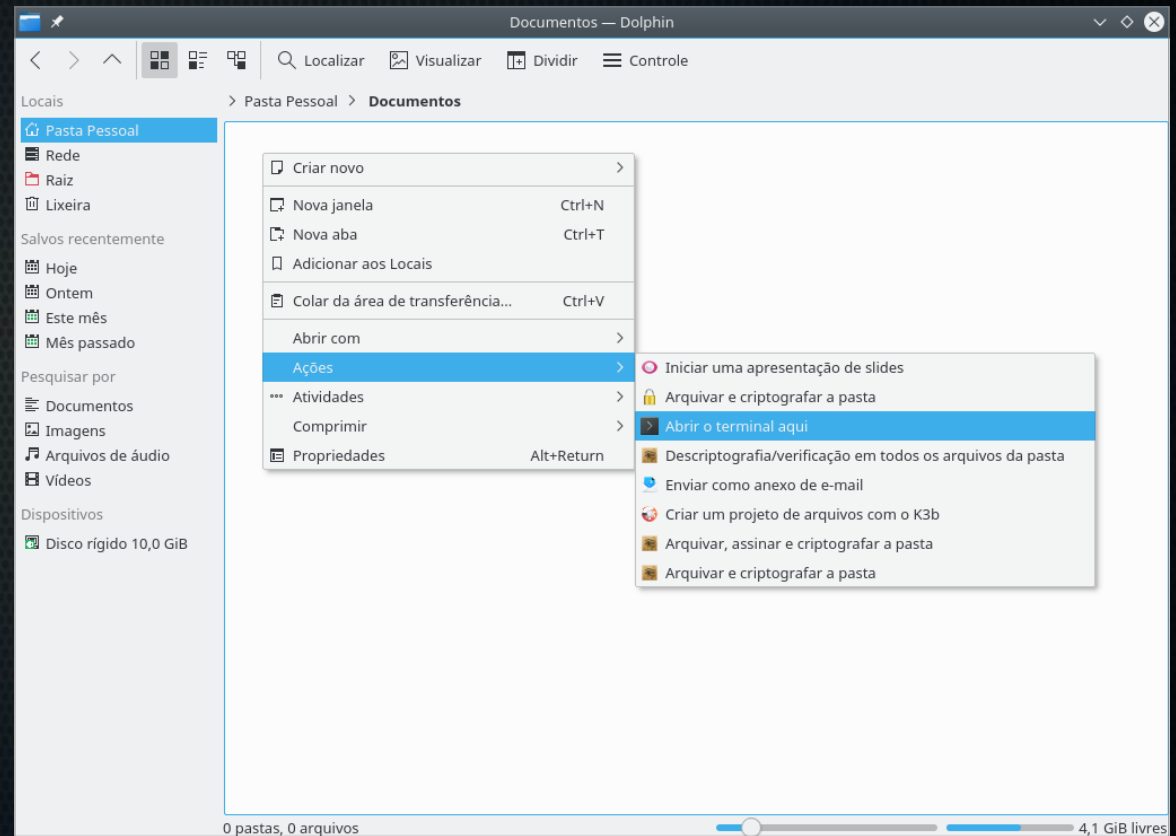
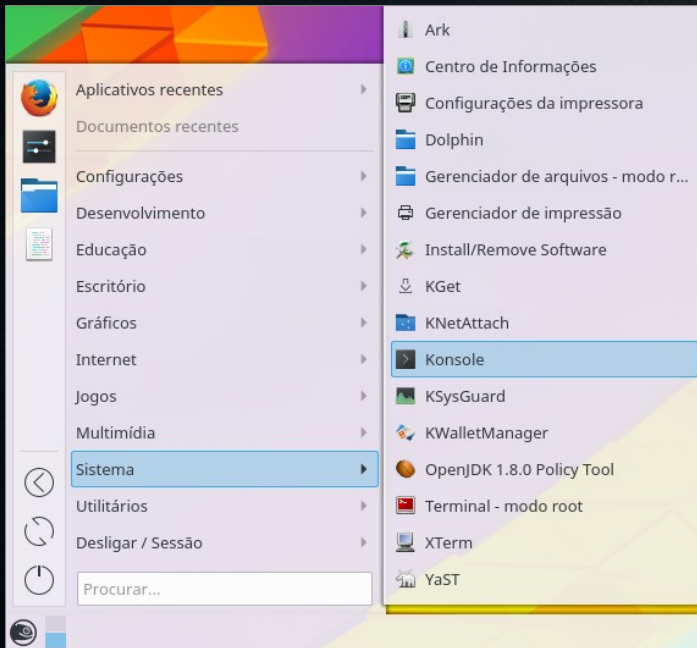
## *Usando o Terminal na GUI*

Um emulador de terminal emula um terminal independente dentro de uma janela na desktop, quer dizer, se comporta como se você acessasse a máquina em um terminal de texto puro sem uma interface gráfica. A maioria dos emuladores de terminal suportam sessões múltiplas, seja em abas ou janelas adicionais.

Por padrão, em ambientes KDE, a aplicação **konsole** é usada para emular um terminal em uma janela. Ele pode ser aberto a partir do submenu Sistema ou inicializado a partir de qualquer pasta usando o navegador Dolphin.

# Operações em Linha de Comando

## *Abrindo o konsole*



# Operações em Linha de Comando

## *Sistema de Janelas X*

A natureza personalizável do Linux te permite derrubar (permanente ou temporariamente) a interface gráfica de janelas X, ou iniciá-la depois que o sistema está executando.

Algumas distribuições Linux distinguem versões de suas mídias de instalação entre desktop (com X) e servidor (geralmente sem). Servidores Linux geralmente são instalados sem X e mesmo se houver, não o lançam durante a inicialização.

Remover o X de um servidor pode ser útil em manter um sistema leve, seguro e de fácil suporte.

# Operações em Linha de Comando

## *Terminais virtuais*

Terminais virtuais (VTs) são sessões de console que usam inteiramente o monitor e teclado fora de um ambiente gráfico. Tais terminais são considerados “virtuais” porque apesar da possibilidade de haver terminais múltiplos ativos, apenas um permanece visível por vez.

Um VT não é a mesma coisa que uma janela de terminal de linha de comando: destas últimas, você pode ter várias visíveis ao mesmo tempo em uma GUI.

# Operações em Linha de Comando

## *Terminais virtuais*

Um terminal virtual (geralmente 1 ou 7) é reservado para o ambiente gráfico, e logins em modo texto são habilitados nos VTs não usados.

Um exemplo de situação em que o uso de um VT é útil é no caso de problemas com a interface gráfica.

Para trocar entre os terminais virtuais, aperte CTRL-ALT-F# (F1 a F12) para o VT correspondente.

# Operações em Linha de Comando

## *Desligando a desktop gráfica*

Distribuições Linux podem iniciar e encerrar a desktop gráfica de várias maneiras. Para sistemas de base RPM (como o openSUSE), o gerenciador de desktop é executado diretamente pelo init com nível de execução 5: mudar para um nível diferente encerra a desktop.

Em um terminal, o comando `sudo telinit 3` pode encerrar a GUI.

# Operações em Linha de Comando

## *Operações básicas: login e logout*

Um terminal de texto disponível te pedirá por um nome de usuário (login:) e senha. Ao digitar a senha, nada aparece no terminal (nem mesmo um \* indicando que você digitou algo) para prevenir que outros vejam sua senha. Ao entrar no sistema, você pode realizar operações básicas.

- Com a sessão iniciada, você pode conectar-se a sistemas remotos via SSH. Por exemplo, digitando `ssh user@server.com`, o SSH conectaria de modo seguro à máquina remota e te daria um terminal de linha de comando, usando senhas ou chaves criptográficas para provar sua identidade.

# Operações em Linha de Comando

## *Operações básicas: Reiniciar e desligar*

O comando `shutdown` é responsável por encerrar devidamente o sistema, apresentando um aviso e então prevenindo o login de outros usuários.

Os comandos `halt` e `poweroff` são equivalentes a `shutdown -h` para desligar o computador. O comando `reboot` equivale a `shutdown -r` e reinicia a máquina. Ambos os casos exigem permissão de superusuário.



# Operações em Linha de Comando

## *Operações básicas: Localizando aplicações*

Dependendo da sua distribuição, programas e pacotes podem ser instalados em variados diretórios. No geral, executáveis devem ficar nos diretórios /bin, /usr/bin, /sbin, /usr/sbin ou sob /opt.

Uma maneira de localizar programas é através das ferramentas *which* e *whereis*. Por exemplo, para encontrar exatamente onde se encontra o programa *diff*:

- \$ *which diff*
- \$ *whereis diff*

# Operações em Linha de Comando

## *Operações básicas: Acessando diretórios*

Ao iniciar um terminal, o diretório padrão deve ser seu diretório home. Você pode imprimir seu exato caminho digitando `echo $HOME`. Os seguintes comandos são úteis para a navegação de diretórios:

- `pwd` : Mostra o diretório atual;
- `cd ~` : Muda ao diretório home;
- `cd ..` : Muda para o diretório acima;
- `cd -` : Muda para o diretório anterior.

# Operações em Linha de Comando

## *Try-it-yourself: Acessando diretórios*

- Localize a aplicação `gcc`;
- Mostre o diretório ativo atual;
- Mude o diretório ativo para `/usr/bin`;
- Mude o diretório ativo para `$HOME`;
- Mude para o diretório imediatamente acima;
- Mude para o diretório anterior;
- Mostre o diretório ativo atual.

# Operações em Linha de Comando

## *Caminhos absolutos e relativos*

Há duas maneiras de se identificar caminhos:

- **Caminho absoluto:** um caminho absoluto começa a partir do diretório / e segue a árvore até alcançar o diretório ou arquivo desejado. Caminhos absolutos sempre começam pela barra.
- **Caminho relativo:** um caminho relativo começa a partir do diretório ativo atual. Caminhos relativos nunca começam pela barra.

# Operações em Linha de Comando

## *Caminhos absolutos e relativos*

Múltiplas barras entre diretórios e arquivos são permitidos, mas apenas uma barra entre elementos do caminho é reconhecida.

`///usr/bin` é válido, mas interpretado pelo sistema como `/usr/bin`.

Na maior parte do tempo é mais conveniente usar caminhos relativos, que requerem menos digitação. Você ainda pode tomar vantagem dos atalhos providenciados por `.` (diretório atual), `..` (diretório acima) e `~` (seu diretório home).

Por exemplo, assuma que você está em seu diretório home e deseja ir ao diretório `/usr/bin`. Os seguintes métodos te levarão a ele:

- Método absoluto: `$ cd /usr/bin`
- Método relativo: `$ cd ../../usr/bin`

# Operações em Linha de Comando

## *Explorando o sistema de arquivos*

Navegar acima e abaixo na árvore do sistema de arquivos pode ser tedioso. O comando `tree` (a ser instalado) é uma boa opção para obter uma visão superior dessa árvore. Use `tree -d` para ver apenas os diretórios e suprimir nomes de arquivos na listagem.

Os seguintes comandos podem ser úteis:

- `cd /` : Muda seu diretório ativo para o diretório raiz;
- `ls` : Lista o conteúdo do diretório ativo;
- `ls -a` : Lista todos os arquivos, incluindo arquivos e diretórios ocultos;
- `tree` : Mostra uma vista em árvore do sistema de arquivos.

# Operações em Linha de Comando

## *Hard Links e Soft Links*

O comando `ln` pode ser usado para criar **hard links** e, com a opção `-s`, criar **soft links** (também chamados **symlinks** - links simbólicos). Esses dois tipos de links são muito úteis em sistemas operacionais baseados em UNIX.

Suponha que um `testel` exista. Um hard link, chamado `teste2`, é criado com o comando

- `$ ln testel teste2`

Isso criará um novo arquivo, mas que sob melhor inspeção (usando o comando `ls -li`) será visível ser um apenas um arquivo com mais de um nome associado, dado seu identificador único.

# Operações em Linha de Comando

## *Hard Links e Soft Links*

Um link simbólico é criado com a opção `-s` como em:

- `$ ln -s teste1 teste3`

Ao inspecionar com o comando `ls -li`, o arquivo `teste3` não aparenta ser um arquivo regular, possui um identificador próprio, e claramente aponta para `teste1`.

Links simbólicos não ocupam espaço adicional em disco. Eles são extremamente convenientes pois podem ser facilmente modificados para apontar para diferentes lugares, inclusive em sistemas de arquivos (ou partições) diferentes que possam ou não estar disponíveis.



# Gerenciando software

## *Sistemas de gerenciamento de pacotes*

As partes centrais de uma distribuição Linux e a maior parte de seu software adicional são instaladas via Sistema de gerenciamento de pacotes.

Cada pacote contém os arquivos e outras instruções necessários para que um componente de software funcione no sistema. Pacotes podem depender uns de outros.

# Gerenciando software

## *Sistemas de gerenciamento de pacotes*

Um sistema de gerenciamento de pacote providencia (ao menos) ferramentas de dois níveis:

- Uma de baixo nível (zypp, rpm), que cuida dos detalhes de desempacotar pacotes individuais, executar scripts, instalar corretamente o software;
- Uma de alto nível (zypper), que trabalha com grupos de pacotes, baixa os pacotes do distribuidor e resolve dependências.

# Gerenciando software

## *Try-it-yourself: Buscar, instalar e remover*

Em um terminal, executar os seguintes comandos (em ordem):

- `$ sudo zypper refresh`
  - Atualizar repositórios
- `$ zypper search lynx`
  - Procurar determinado pacote
- `$ sudo zypper install tree lynx (confirmar ações)`
  - Instalar software
- `$ zypper info lynx`
  - Verificar o status do pacote
- `$ sudo zypper remove lynx (confirmar ações)`
  - Remover software
- `$ sudo zypper update`
  - Atualizar software já instalado