

[VPS](#) [jul 21, 2023](#) [Ariane G.](#) 18min de leitura

40 Comandos Linux Que Todo Usuário Deve Conhecer



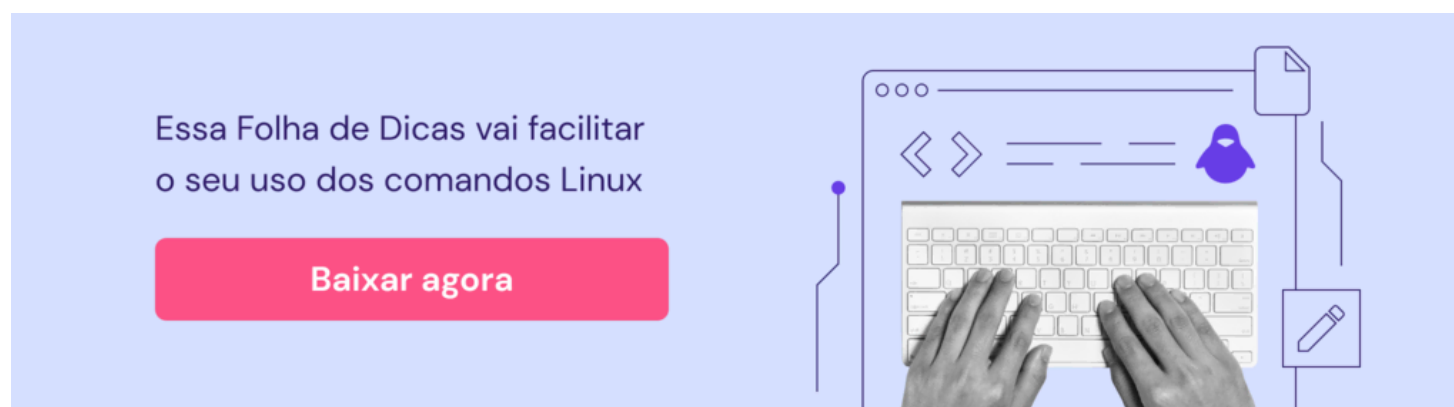
Ao ouvir falar do Linux, a maioria das pessoas acha que ele é um sistema operacional complicado e que apenas programadores conseguem usar. Na realidade, entretanto, não é nada disso!

O Linux é uma família de sistemas operacionais Unix de código aberto, baseados no kernel Linux. A família inclui versões muito famosas, como Ubuntu, Fedora, Mint, Debian e outros. No universo do Linux, cada uma dessas versões é chamada de distribuição.

Ao operar um sistema operacional Linux, você precisa usar um **shell**, que é uma interface que fornece acessos aos serviços do sistema operacional. A maioria da distribuições Linux implementam uma interface gráfica do usuário (GUI) como shell, principalmente para fornecer facilidade de uso para seus usuários.

Dito isto, dependendo da sua tarefa, é recomendável usar uma [interface de linha de comando \(CLI\)](#). Isso porque uma mesma tarefa que exigirá várias etapas repetitivas na GUI poderá ser realizada com apenas poucos comandos (e em poucos segundos) na CLI.

Portanto, se você estiver pensando em usar o Linux no seu projeto, aprender as linhas de comando básicas do sistema ajudará bastante. Neste artigo, você vai aprender 40 dos comandos Linux mais importantes que vão, indiscutivelmente, ajudar você a navegar pelo Linux e usar o sistema operacional sem problemas no seu [servidor VPS](#) — mesmo que você seja um iniciante!




Conteúdo

- Tutorial em Vídeo >
- O Que É Um Comando Linux? >
- Comandos Mais Usados no Linux >
- Lista Os 40 Principais Comandos Linux >
- Dicas e Truques Bônus >
- Comandos Linux Perguntas Frequentes >

Tutorial em Vídeo

No vídeo abaixo, você pode conferir alguns dos 15 comandos Linux mais úteis para iniciantes.

15 Comandos Linux Para Iniciantes - Hostinger Brasil



O Que É Um Comando Linux?

Um comando Linux é um programa que você executa diretamente na linha de comando (CLI). Isto é, você digita o comando na CLI e, em seguida, a interface processa aquela instrução e realiza o comando solicitado.

No Linux, os comandos são executados no Terminal — basta pressionar **Enter** ao terminar de digitar (ou copiar) o comando desejado. Você pode executar comandos para realizar uma série de tarefas, como instalação de pacotes, gerenciamento de usuários e manipulação de arquivos.

As três partes principais de um comando são:

- **CommandName (nome).** É a regra em si que você quer executar.
- **Flag (opção).** É um modificador para a operação do comando. Você pode incluí-lo no comando usando um hífen (-) ou dois (--).
- **Argument (parâmetro).** Serve para adicionar informações ou contexto ao comando.

Tendo isso em mente, a sintaxe básica de um comando Linux é a seguinte:

```
NomeDoComando [opção/opções] [parâmetro(s)]
```

Comandos Mais Usados no Linux

Confira abaixo uma lista com os 8 comandos mais usados nos sistemas operacionais baseados em Linux:

1. **pwd**: encontra o caminho completo do diretório atual.
2. **cd**: permite navegar até determinada pasta.
3. **ls**: lista todos os arquivos e pastas dentro de um diretório.
4. **cat**: lista os conteúdos de um arquivo de texto na saída padrão (stdout).
5. **cp**: copia arquivos do diretório atual para uma pasta diferente.
6. **mv**: pode ser usado para mover ou renomear arquivos.
7. **mkdir**: cria um novo diretório.
8. **rm**: remove arquivos e diretórios.
9. **sudo**: executa um comando como superusuário.
10. **find**: para buscar arquivos em diretórios.



Lista: Os 40 Principais Comandos Linux

Antes de entrarmos na lista de principais comandos Linux, primeiro você precisa abrir a linha de comando do sistema. Se você ainda tem insegurança sobre a interface de linha de comando, confira nosso [tutorial sobre CLI](#).

Embora os passos abaixo possam ser ligeiramente diferentes dependendo da distribuição que você estiver usando, você geralmente encontra a linha de comando na seção **Utilities** (Utilidades).

Vamos começar?

1. Comando pwd

Use o comando **pwd** para encontrar o caminho para o diretório atual (da pasta) em que você está. O comando vai retornar um caminho completo (cheio), que é basicamente um caminho que começa com uma barra inclinada (**/**). Um exemplo de um caminho completo é **/home/username**.

O comando **pwd** usa a seguinte sintaxe:

pwd [opção]

Ele tem duas opções aceitáveis:

- **-L** ou **-logical** imprime o conteúdo da variável de ambiente, incluindo links simbólicos.
- **-P** ou **-physical** imprime o caminho real do diretório atual.

2. Comando cd

Para navegar pelos arquivos e diretórios Linux, use o comando **cd**. Ele requer ou um caminho completo ou o nome de um diretório, dependendo do diretório atual em que você estiver.

Vamos dizer que você esteja em **/home/username/Documents** e quer ir para **Photos**, um subdiretório de

`/nome/username/movies`. Nesse caso, você tem que digitar **cd** seguido pelo caminho absoluto do diretório.

Existem alguns atalhos que você pode usar para navegar mais rapidamente.:

- Use **cd ..** (com dois pontos seguidos) para subir diretório acima
- Use **cd ~[username]** para acessar o diretório inicial de outro usuário.
- Use **cd-** (com um hífen) para mover para os diretórios anteriores.

3. Comando ls

O comando **ls** é usado para visualizar conteúdos em um diretório. Por padrão, esse comando vai mostrar os conteúdos apenas do diretório atual em que você estiver.

Se você quiser ver o conteúdo de outros diretórios, digite **ls** e, então, o caminho do diretório. Por exemplo, digite **ls /home/username/Documents** para ver os conteúdos de **Documents**.

Existem variações que você pode usar com o comando **ls**:

- **ls -R** vai listar todos os arquivos nos subdiretórios.
- **ls -a** vai mostrar todos os arquivos ocultos.
- **ls -lh** vai listar todos os tamanhos de arquivos em formatos fáceis, como MB, GB ou TB.

4. Comando cat

O **cat** (abreviação para concatenar) é um dos comandos Linux mais usados. Ele é usado para visualizar, criar e relacionar arquivos. Para executar esse comando, digite **cat** seguido pelo nome do arquivo e sua extensão. Por exemplo: **cat nomedoarquivo.txt**.

Aqui estão outras maneiras de usar o [comando cat](#):

- **cat > nomedoarquivo.txt** cria um novo arquivo
- **cat nomedoarquivo1.txt nomedoarquivo2.txt > nomedoarquivo3.txt** junta dois arquivos (1 e 2) em um novo (3).
- **tac nomedoarquivo.txt** exibe o conteúdo do arquivo em ordem reversa.

5. Comando cp

Use o comando **cp** para copiar arquivos ou diretórios e seu conteúdo. Abaixo, listamos alguns exemplos.

Para copiar um arquivo do diretório atual para outro, digite **cp** seguido do nome do arquivo e do diretório de destino. Por exemplo:

cp nomedoarquivo.txt /home/username/Documents

Para copiar arquivos para um diretório, digite os nomes dos arquivos seguidos do diretório de destino:

cp nomedoarquivo1.txt nomedoarquivo2.txt nomedoarquivo3.txt /home/username/Documents

Para copiar o conteúdo de um arquivo para um novo arquivo no mesmo diretório, digite **cp** seguido do arquivo de origem e do arquivo de destino:

cp nomedoarquivo1.txt nomedoarquivo2.txt

Para copiar um diretório inteiro, passe o sinalizador **-R** antes de digitar o diretório de origem, seguido pelo diretório de destino:

cp -R /home/username/Documents /home/username/Documents_backup

6. Comando mv

O uso mais comum do comando **mv** é mover arquivos, mas ele também pode ser usado para renomear arquivos.

Basta digitar **mv** seguido do nome do arquivo e do diretório de destino. Por exemplo, você deseja mover o arquivo **nomedoarquivo.txt** para o diretório **/home/username/Documents**:

```
mv nomedoarquivo_antigo.txt nome_novo.txt
```

7. Comando mkdir

Use o comando **mkdir** para criar um ou vários diretórios de uma só vez e definir permissões para cada um deles. O usuário que executa esse comando deve ter o privilégio de criar uma nova pasta no diretório principal, caso contrário, poderá receber um erro de permissão negada.

Aqui está a sintaxe básica:

```
mkdir [opção] nome_do_diretório
```

Por exemplo, você deseja criar um diretório chamado **Music**:

```
mkdir Music
```

Para criar um novo diretório chamado **Songs** dentro de **Music**, use este comando:

```
mkdir Music/Songs
```

O comando **mkdir** aceita muitas opções, como:

- **-p** ou **-parents** cria um diretório entre duas pastas existentes. Por exemplo, **mkdir -p Music/2020/Songs** criará o novo diretório “2020”.
- **-m** define as permissões do arquivo. Por exemplo, para criar um diretório com permissões completas de leitura, gravação e execução para todos os usuários, digite **mkdir -m777 nome_do_diretório**.
- **-v** imprime uma mensagem para cada diretório criado.

8. Comando rmdir

Para excluir permanentemente um diretório vazio, use o [comando rmdir](#). Lembre-se de que o usuário que executa esse comando deve ter privilégios **sudo** no diretório pai.

Por exemplo, você deseja remover um subdiretório vazio chamado **personal1** e sua pasta principal **mydir**:

```
rmdir -p mydir/personal1
```

9. Comando rm

O comando **rm** é usado para excluir arquivos em um diretório. Certifique-se de que o usuário que executa esse comando tenha permissões de gravação.

Lembre-se do local do diretório, pois isso apagará o(s) arquivo(s) permanentemente e não há como desfazer a ação.

Aqui está a sintaxe geral:

```
rm nome_do_arquivo
```

Para remover vários arquivos, digite o seguinte comando:

```
rm nome_do_arquivo1 nome_do_arquivo2 nome_do_arquivo3
```

Aqui estão algumas opções que você pode adicionar:

- **-i** solicita a confirmação do sistema antes de excluir um arquivo.
- **-f** permite que o sistema faça a remoção sem confirmação.
- **-r** exclui arquivos e diretórios recursivamente.

10. Comando touch

O [comando touch](#) permite criar um arquivo vazio ou gerar e modificar um registro de data e hora na linha de comando do Linux.

Por exemplo, digite o seguinte comando para criar um arquivo HTML chamado **Web** no diretório

11. Comando locate

Você pode o [comando locate](#) para localizar um arquivo, assim como você faz para procurar um arquivo no Windows. Além disso, usando o argumento **-i** junto com esse comando faz com que ele se torne insensível a maiúsculas ou minúsculas, permitindo que você pesquise por um arquivo mesmo sem saber exatamente o nome dele.

Para procurar um arquivo que contém duas ou mais palavras, use um asterisco (*). Por exemplo, use o comando **locate -i school*note** para encontrar qualquer arquivo que tenha as palavras “school” e “note”, não importando se existem letras maiúsculas ou minúsculas.

12. Comando find

Similar ao comando **locate**, o comando **find** ajuda você a procurar por arquivos. A diferença é que você usa o **find** para localizar arquivos dentro de um diretório específico.

Como exemplo, digite **find /home/ -name notes.txt** para procurar por um arquivo chamado **notes.txt** dentro do diretório **home** e seus subdiretórios.

Outras variações na hora de usar o **find** são:

- **find -name nomedoarquivo.txt** para localizar arquivos no diretório atual.
- **find ./ -type d -name nomedodiretorio** para procurar diretórios.

13. Comando grep

Outro comando básico do Linux que merece ser citado é o **grep**. Ele permite que você encontre uma palavra pesquisando todo o conteúdo de um arquivo específico.

Quando o [comando grep](#) encontra uma correspondência, ele imprime todas as linhas que contêm o padrão específico. Esse comando ajuda a filtrar arquivos de registro grandes.

Por exemplo, se você deseja pesquisar a palavra **blue (azul)** no arquivo **notepad.txt**:

```
grep blue notepad.txt
```

A saída do comando exibirá as linhas que contêm a palavra **blue**.

14. Comando sudo

Correspondente a **SuperUser Do**, **sudo** é um dos comandos básicos mais populares do Linux. Ele permite executar tarefas que exigem permissões administrativas ou de root.

Ao usar o **sudo**, o sistema solicitará que os usuários se autenticuem com uma senha. Em seguida, o sistema Linux registrará um registro de data e hora como um rastreador. Por padrão, todo usuário root pode executar comandos **sudo** por **15 minutos/sessão**.

Se você tentar executar o **sudo** na linha de comando sem se autenticar, o sistema registrará a atividade como um evento de segurança.

Aqui está a sintaxe geral:

```
sudo [comando]
```

Você também pode adicionar uma opção, por exemplo:

- **-k** ou **–reset-timestamp** invalida o arquivo de registro de data e hora.
- **-g** ou **–group=group** executa comandos como um nome ou ID de grupo especificado.
- **-h** ou **–host=host** executa comandos no host.

15. Comando df

Use o [comando df](#) para obter informações sobre o uso do espaço em disco do sistema, mostrado em porcentagem e quilobyte (KB). Esta é a sintaxe geral:

regivei por numanos:

df -h

Essas são algumas opções que você pode usar:

- O **df -m** exibe informações sobre o uso do sistema de arquivos em **MBs**.
- **df -k** exibe o uso do sistema de arquivos em **KBs**.
- **df -T** mostra o **tipo** de sistema de arquivos em uma nova coluna.

16. Comando du

Se você quiser verificar quanto de espaço um arquivo ou diretório ocupa, use o comando **du**. Você pode executar esse comando para identificar qual parte do sistema usa excessivamente o armazenamento do seu sistema.

Lembre-se de que você deve especificar o caminho do diretório ao usar o comando **du**. Por exemplo, para verificar **/home/user/Documents**, digite:

du /home/user/Documents

Adicionar um sinalizador ao comando **du** modificará a operação, por exemplo:

- **-s** oferece o tamanho total de uma pasta especificada.
- **-m** fornece informações sobre pastas e arquivos em **MB**
- **k** exibe informações em **KB**.
- **-h** informa a data da última modificação das pastas e arquivos exibidos.

17. Comando head

O comando **head** permite que você visualize as primeiras dez linhas de um texto. A adição de uma opção permite que você altere o número de linhas mostradas. O comando **head** também é usado para enviar dados canalizados para a CLI.

Aqui está a sintaxe geral:

head [opção] [arquivo]

Por exemplo, se você deseja visualizar as primeiras dez linhas do arquivo **note.txt**, localizado no diretório atual:

head note.txt

Abaixo estão algumas opções que você pode adicionar:

- **-n** ou **–lines** imprime o primeiro número personalizado de linhas. Por exemplo, digite **head -n 5 nomedoarquivo.txt** para exibir as cinco primeiras linhas de **nomedoarquivo.txt**.
- **-c** ou **–bytes** imprime o primeiro número personalizado de bytes de cada arquivo.
- **-q** ou **–quiet** não imprimirá cabeçalhos que especifiquem o nome do arquivo.

18. Comando tail

O [comando tail](#) tem função similar ao comando **head**, mas mostrando as últimas 10 linhas de um arquivo de texto.

Este é o formato geral:

tail [opção] [arquivo]

Por exemplo, você deseja mostrar as últimas dez linhas do arquivo **colors.txt**:

tail -n colors.txt

19. Comando diff

O comando **diff** (diferença) compara o conteúdo de dois arquivos linha por linha. Depois de analisar esses

programas. Assim, eles não precisam reescrever o código inteiro.

Este é o formato geral:

diff [opção] arquivo1 arquivo2

Por exemplo, você deseja comparar dois arquivos de texto — **note.txt** e **note_update.txt**:

diff note.txt note_update.txt

Aqui estão algumas opções para adicionar:

- **-q** mostra apenas se os arquivos são diferentes ou não, sem especificar as diferenças.
- **-i** torna o comando **diff** insensível a maiúsculas e minúsculas.
- **-b** ignora espaços em branco como possíveis diferenças.

20. Comando tar

O [comando tar](#) reúne vários arquivos em um arquivo **TAR** — um formato do Linux semelhante ao **ZIP**, com compactação opcional.

Aqui está a sintaxe básica:

tar [options] [archive_file] [arquivo ou diretório a ser arquivado]

Por exemplo, você deseja criar um novo arquivo **TAR** chamado **novoarquivo.tar** no diretório **/home/user/Documents**:

tar -cvf novoarquivo.tar /home/user/Documents

O comando **tar** aceita muitas opções, como:

- **-x** extrai um arquivo.
- **-t** lista o conteúdo de um arquivo.
- **-u** arquivar e adicionar a um arquivo existente.

21. Comando chmod

O **chmod** é um comando que modifica as permissões de leitura, gravação e execução de um arquivo ou diretório. No Linux, cada arquivo está associado a três classes de usuários: **proprietário**, **membro do grupo** e **outros**.

Aqui está a sintaxe básica:

chmod [opção] [permissão] [nome_do_arquivo]

Por exemplo, o proprietário é atualmente o único com permissões totais para alterar o arquivo **note.txt**. Para permitir que os membros do grupo e outros possam ler, gravar e executar o arquivo, altere-o para o tipo de permissão **-rwxrwxrwx**, cujo valor numérico é **777**:

chmod 777 note.txt

O [comando chmod](#) oferece suporte a várias opções, incluindo:

- **-c** ou **–changes** exibe informações quando uma alteração é feita.
- **-f** ou **–silent** suprime as mensagens de erro.
- **-v** ou **–verbose** exibe um diagnóstico para cada arquivo processado.

22. Comando chown

No Linux, todos os arquivos são de propriedade de um usuário específico. O [comando chown](#) permite alterar a propriedade de um arquivo, diretório ou link simbólico para um nome de usuário específico.

Este é o formato básico:

chown [opção] proprietário[:grupo] arquivo(s)

23. Comando jobs

Um job é um processo que iniciado pelo shell. O comando **jobs** exibirá todos os processos em execução juntamente com seus status. Lembre-se de que esse comando só está disponível nos shells **cs**h, **ba**sh, **tc**sh e **k**sh.

Essa é a sintaxe básica:

jobs [opções] jobID

Para verificar o status dos trabalhos no shell atual, basta digitar **jobs** na CLI.

Aqui estão algumas opções que você pode usar:

- **-l** lista as IDs de processo juntamente com suas informações.
- **-n** lista os processos cujos status foram alterados desde a última notificação.
- **-p** lista somente IDs de processos.

24. Comando kill

Se você tem um programa que não está respondendo bem, você pode finalizá-lo manualmente pelo [comando kill](#). Ele vai mandar um certo sinal ao aplicativo com mau funcionamento e instruir que ele seja encerrado sozinho logo na sequência.

Existe um total de [64 avisos](#) que você pode usar, mas, geralmente, as pessoas usam apenas 2 deles:

- **SIGTERM (15)** – pede que um programa pare de rodar e dá algum tempo para salvar todo o seu progresso. Se você não especificar o aviso quando executar o comando kill, é este aviso que será usado.
- **SIGKILL (9)** – força um programa a parar imediatamente, em que todo o progresso não salvo será perdido.

Além de saber os avisos (sinais, notificações), você também precisa conhecer o número de identificação do processo (PID) do programa que você quer **matar** (kill). Se você não souber o PID, apenas execute o comando **ps ux**.

Depois de saber qual aviso você quer usar e o PID do programa, use a sintaxe abaixo:

kill [signal option] PID.

25. Comando ping

O [comando ping](#) é um dos comandos básicos do Linux mais usados para verificar se uma rede ou um servidor está acessível. Além disso, ele é usado para solucionar vários problemas de conectividade.

Este é o formato geral:

ping [opção] [nome_do_hospedeiro_ou_endereço_IP]

Por exemplo, você quer saber se pode se conectar ao **Google** e medir seu tempo de resposta:

ping google.com

26. Comando wget

A linha de comando do Linux permite que você baixe arquivos da Internet usando o [comando wget](#). Ele funciona em segundo plano, sem atrapalhar outros processos em execução.

O comando **wget** baixa arquivos usando os protocolos HTTP, HTTPS e FTP. Ele pode executar downloads recursivos, que transferem partes de sites seguindo estruturas de diretórios e links, criando versões locais de páginas da web.

Para usá-lo, digite o seguinte comando:

`wget https://wordpress.org/latest.zip`

27. Comando `uname`

O comando **uname**, que significa Unix Name, vai mostrar informações detalhadas sobre seu sistema Linux. Isso inclui o nome da máquina, do sistema operacional, do kernel e assim por diante.

uname [opção]

Essas são algumas das opções que você pode usar:

- **-a** exibe todas as informações do sistema.
- **-s** exibe o nome do kernel.
- **-n** exibe o hostname do node do sistema.

28. Comando `top`

Equivalente ao gerenciador de tarefas do Windows, o [comando top](#) vai mostrar uma lista de processos que estão em execução e o quanto de CPU cada processo usa. Ele é muito útil para monitorar o uso dos recursos do sistema, especialmente para saber qual processo deve ser encerrado por consumir muitos recursos. Basta digitar **top** na CLI para executá-lo.

29. Comando `history`

Com o **history**, o sistema listará até 500 comandos executados anteriormente, permitindo que você os reutilize sem precisar digitá-los novamente. Lembre-se de que somente os usuários com privilégios **sudo** podem executar esse comando. A forma de execução desse comando também depende do shell do Linux que você usa.

Para executá-lo, digite o comando abaixo:

history [opção]

Esse comando oferece suporte a várias opções, como:

- **-c** limpa a lista completa do histórico.
- **-d offset** exclui o registro do histórico na posição **OFFSET**.
- **-a** acrescenta linhas de histórico.

30. Comando `man`

O comando **man** fornece um manual completo para todos comandos ou utilitários que podem ser executados no Terminal, incluindo o nome, a descrição e as opções.

Ele consiste em nove seções:

- Programas executáveis ou comandos do shell
- Chamadas de sistema
- Chamadas da biblioteca
- Jogos
- Arquivos especiais
- Formatos e convenções de arquivos
- Comandos de administração do sistema
- Rotinas do kernel
- Diversos

Para exibir o manual completo, digite:

man [nome_do_comando]

Por exemplo, se você deseja acessar o manual do comando **ls**:

man [opção] [número_da_seção] [nome_do_comando]

Por exemplo, você deseja ver a seção 2 do manual do comando **ls**:

man 2 ls

31. Comando echo

O comando **echo** é um utilitário nativo que exibe uma linha de texto ou cadeia de caracteres (string) usando a saída padrão. Veja a seguir a sintaxe básica:

echo [opção] [string]

Por exemplo, você pode exibir o texto **Hostinger Tutoriais** digitando:

echo "Hostinger Tutoriais"

Esse comando oferece suporte a várias opções, como:

- **-n** exibe a saída sem a nova linha final.
- **-e** permite a interpretação dos seguintes escapes de barra invertida:
 - **\a** reproduz um alerta sonoro.
 - **\b** remove os espaços entre um texto.
 - **\c** não produz mais nenhum resultado.
- **-E** exibe a opção padrão e desativa a interpretação dos escapes de barra invertida.

32. Comandos zip e unzip

Use o comando **zip** para compactar seus arquivos em um arquivo **ZIP**, um formato universal comumente usado no Linux. Ele pode escolher automaticamente a melhor taxa de compactação.

O comando **zip** também é útil para arquivar arquivos e diretórios e reduzir o uso do disco.

Para usá-lo, digite a seguinte sintaxe:

zip [opções] zipfile file1 file2....

Por exemplo, se você tem um arquivo chamado **note.txt** que deseja compactar em **archive.zip** no diretório atual:

zip archive.zip note.txt

Por outro lado, o [comando unzip](#) extrai os arquivos compactados de um arquivo. Este é o formato geral:

unzip [opção] nome_do_arquivo.zip

Portanto, para descompactar um arquivo chamado **archive.zip** no diretório atual, digite:

unzip archive.zip

33. Comando hostname

Execute o comando **hostname** para saber o nome do host do sistema. Você pode executá-lo com ou sem uma opção. Aqui está a sintaxe geral:

hostname [opção]

Há muitos sinalizadores opcionais a serem usados, incluindo:

- **-a** ou **-alias** exibe o alias do nome do host.
- **-A** ou **-all-fqdns** exibe o nome de domínio totalmente qualificado (FQDN) da máquina.
- **-i** ou **-ip-address** exibe o endereço IP da máquina.

Por exemplo, digite o seguinte comando para saber o endereço IP de seu computador:

34. Comandos useradd e userdel

O Linux é um sistema multiusuário, o que significa que mais de uma pessoa pode usá-lo simultaneamente. **useradd** é usado para criar uma nova conta, enquanto o comando **passwd** permite adicionar uma senha. Somente aqueles com privilégios de root ou **sudo** podem executar o comando **useradd**.

Quando você usa o comando **useradd**, ele realiza algumas alterações importantes:

- Edita os arquivos **/etc/passwd**, **/etc/shadow**, **/etc/group** e **/etc/gshadow** para as contas recém-criadas.
- Cria e preenche um diretório inicial para o usuário.
- Define as permissões e a propriedade dos arquivos no diretório inicial.

Aqui está a sintaxe básica:

useradd [opção] nome_de_usuario

Para definir a senha:

passwd a_senha_escolhida

Por exemplo, para adicionar uma nova pessoa chamada Paulo, digite o seguinte comando simultaneamente:

useradd Paulo

passwd 123456789

Para excluir uma conta de usuário, use o comando **userdel**:

userdel nome_de_usuario

35. Comando apt-get

O **apt-get** é uma ferramenta de linha de comando para lidar com as bibliotecas da Advanced Package Tool (APT) no Linux. Ele permite que você obtenha informações e pacotes de fontes autenticadas para gerenciar, atualizar, remover e instalar softwares e suas dependências.

A execução do comando **apt-get** exige que você tenha privilégios **sudo** ou root.

Aqui está a sintaxe principal:

apt-get [opções] (comando)

Esses são os comandos mais comuns que você pode adicionar ao **apt-get**:

- **update** sincroniza os arquivos de pacote de suas fontes.
- **upgrade** instala a versão mais recente de todos os pacotes instalados.
- **check** atualiza o cache de pacotes e verifica dependências quebradas.

36. comandos nano, vi e jed

O Linux permite que os usuários editem e gerenciem arquivos por meio de um editor de texto usando comandos como o **nano**, o **vi** ou o **jed**. O **nano** e o **vi** são nativos do sistema operacional, enquanto o **jed** precisa ser instalado.

O **comando nano** denota palavras-chave e pode funcionar com a maioria dos idiomas. Para usá-lo, digite o seguinte comando:

nano [nome do arquivo]

O **vi** usa dois modos operacionais para trabalhar: **insert** e **Command**. O **insert** é usado para editar e criar um arquivo de texto. Por outro lado, o **command** executa operações, como salvar, abrir, copiar e colar um arquivo.

Para usar o **vi** em um arquivo, digite:

vi [nome do arquivo]

Para abrir o programa, basta digitar **jed** na linha de comando.

37. Comandos alias e unalias

O **alias** permite que você crie um atalho com a mesma funcionalidade de um comando, nome de arquivo ou texto. Quando executado, ele instrui o shell a substituir uma string por outra.

Para usar o comando **alias**, digite a seguinte sintaxe:

alias Name=String

Por exemplo, você deseja tornar **k** o alias do comando **kill**:

alias k='kill'

Por outro lado, o comando **unalias** exclui um alias existente.

Veja a seguir como é a sintaxe geral:

unalias [nome_do_alias]

38. Comando su

O comando switch user, ou **su**, permite executar um programa como um usuário diferente. Ele altera a conta administrativa na sessão de login atual. Esse comando é especialmente útil para acessar o sistema por meio de [SSH](#) ou usar o gerenciador de exibição da GUI quando o usuário raiz não está disponível.

Esta é a sintaxe geral do comando:

su [opções] [nome de usuário [argumento]]

Quando executado sem nenhuma opção ou argumento, o comando **su** é executado com privilégios de root. Ele solicitará que você se autentique e use os privilégios **sudo** temporariamente.

Aqui estão algumas que você pode usar:

- **-p** ou **–preserve-environment** mantém o mesmo ambiente de shell, composto por HOME, SHELL, USER e LOGNAME.
- **-s** ou **–shell** permite especificar um ambiente de shell diferente para execução.
- **-l** ou **–login** executa um script de login para mudar para um nome de usuário diferente. Para executá-lo, é necessário digitar a senha do usuário.

39. Comando htop

O comando **htop** é um programa interativo que monitora os recursos do sistema e os processos do servidor em tempo real. Ele está disponível na maioria das distribuições Linux e você pode instalá-lo usando o gerenciador de pacotes padrão.

Em comparação com o comando **top**, o **htop** tem muitos aprimoramentos e recursos adicionais, como a operação com o mouse e indicadores visuais.

Para usá-lo, execute o seguinte comando:

htop [opções]

Você também pode adicionar opções, como:

- **-d** ou **–delay** mostra o atraso entre as atualizações em décimos de segundos.
- **-C** ou **–no-color** ativa o modo monocromático.
- **-h** ou **–help** exibe a mensagem de ajuda e sai.

40. Comando ps

O status do processo, ou comando **ps**, produz um snapshot de todos os processos em execução em seu sistema. Os resultados estáticos são obtidos dos arquivos virtuais no sistema de arquivos **/proc**.

- A ID exclusiva do processo (**PID**)
- O tipo de terminal (**TTY**)
- O tempo de execução (**TIME**)
- O comando que inicia o processo (**CMD**)

Aqui estão algumas opções que você pode usar:

- **-T** exibe todos os processos associados à sessão atual do shell.
- **-u nome_de_usuario** lista os processos associados a um usuário específico.
- **-A** ou **-e** mostra todos os processos em execução.



Dicas e Truques Bônus

Aqui estão algumas dicas e truques que você pode usar para gerenciar o sistema Linux:

- Digite o comando **clear** para limpar a tela do Terminal.
- Pressione o botão **Tab** para ativar o preenchimento automático depois de digitar um comando com um argumento.
- Use **Ctrl + C** para encerrar um comando em execução.
- Pressione **Ctrl + Z** para pausar um comando em funcionamento.
- Use **Ctrl + S** para congelar seu Terminal temporariamente.
- Pressione **Ctrl + Q** para desfazer o congelamento do Terminal.
- Use **Ctrl + A** para ir para o início da linha.
- Pressione **Ctrl + E** para ir até o final da linha.
- Ao executar vários comandos em uma única linha, use **(;)** para separá-los. Como alternativa, use **&&** para permitir que o próximo comando seja executado somente se o anterior for bem-sucedido.

Conclusão

Os comandos do Linux permitem que você execute tarefas básicas e administrativas de forma rápida e eficaz no Terminal. Com ele, você terá mais flexibilidade sobre o seu sistema e a capacidade de acessar recursos que nem sempre estão disponíveis por meio de uma interface gráfica.

Aprender os comandos básicos do Linux é muito importante para interagir com sua máquina, principalmente se você usa um [servidor virtual privado](#) (VPS). Na maioria dos casos, cada comando consiste em três partes: um **nome de comando**, um **sinlizador** ou **opção** e um **parâmetro** ou **argumento**.

Este artigo abordou 40 comandos comuns, como o **apt-get** para instalar um pacote, o **nano** para manipular um arquivo, o **htop** para monitorar os processos atuais e o **ls** para visualizar um diretório.

Esperamos que este artigo tenha ajudado você a aprender os comandos básicos do Linux. Se tiver alguma dúvida ou sugestão, deixe-a na seção de comentários abaixo.

Até mais!

Outras Dicas de Linux Para Aprender e Salvar

[Como usar o comando Time no Linux](#)[Como usar o comando Scp para transferência de arquivos](#)[Como usar o comando Watch no Linux](#)[Como usar o comando Shutdown](#)[Como usar o comando Tee para gravar e exibir em um arquivo](#)[Como localizar o tipo de arquivo com o comando File no Linux](#)[Como listar serviços no Linux](#)[Como usar o Rsync para transferência e sincronização de arquivos](#)[Como alterar uma senha de usuário no Linux](#)[Como criar um link simbólico com o comando ln](#)

Comandos Linux: Perguntas Frequentes

Confira abaixo algumas das perguntas mais frequentes feitas por usuários que estão aprendendo sobre comandos Linux.

Qual o Comando Linux Mais Básico?

Um dos comandos Linux mais comumente utilizados é o comando cat (abreviação de concatenar). Este comando ajuda a criar, visualizar e combinar arquivos, além de redirecionar a saída (isto é, o conteúdo de todo o arquivo).

Quantos Comandos Existem no Linux?

Existem mais de 100 comandos compartilhados pelo Linux e outros sistemas operacionais similares.

O AUTOR

Ariane G.

A Ariane é SEO Team Leader com experiência em conteúdo, localização e SEO. Seu desafio é levar a Hostinger ao topo dos resultados no Brasil, França e Reino Unido. Eventualmente ela compartilha seu conhecimento no blog e nos tutoriais da Hostinger e no tempo livre ela gosta de explorar lugares novos e viver intencionalmente.

[Mais de Ariane G.](#)



08 nov • [VPS](#)

[Como Hospedar Bot Para Discord: Saiba Criar Seu Próprio Bot no VPS em 2023](#)

Adicionar bots ao seu servidor Discord permite automatizar tarefas e agregar novos recursos ao ambiente. Embora existam muitos bots populares do...

[Por Bruno Santana](#)

17 out • [VPS](#)

[O Que é o Arquivo PHP.ini e Onde ele Fica? Entenda os Principais Parâmetros e Como Editá-lo](#)

O PHP.ini é um arquivo de configuração que contém as definições de PHP do seu servidor web. Sempre que o PHP for iniciado, seu sistema...

[Por Carlos E.](#)

29 set • [VPS](#)

[Docker Cheat Sheet: Guia de Referência com Comandos Básicos – Códigos Prontos para Usar](#)

Docker é uma plataforma de código-aberto muito popular, utilizada para facilitar os processos de desenvolvimento, teste, lançamento e gerenciamento...

[Por Bruna B. Barro](#)

Excelente

Baseado em 21.946 avaliações

Comentários

[Faça um comentário](#)**jurandir**

agosto 30 2019

[RESPONDER](#)

boa noite, acabei de instalar debian 10, sozinho, na instalacao pede senha de usuario e senha de root, apoz a instalacao, vou usar terminal como root, nao aceita a senha de root, o que faco?

Ariane G.

setembro 03 2019

Oi, Jurandir. Você pode chamar nossa equipe de suporte ao cliente pelo seu painel de controle. Nosso time de experts está pronto para ajudar! ;)

lury

outubro 14 2022

Você conseguiu resolver o problema? Estou com o mesmo problema rs

Carlos E.

outubro 21 2022

Olá, luri! Falei com a nossa equipe técnica e eles recomendaram que você redefina a sua senha root! A senha SSH é a mesma coisa que a senha root. Então se ela estiver incorreta por algum motivo, você pode redefini-la a qualquer momento

**Renato Lanzieri**

agosto 01 2021

[RESPONDER](#)

Excelente!!

João Martins

agosto 27 2021

[RESPONDER](#)

Wesley

setembro 06 2021

[RESPONDER](#)

Lista muito boa ;) Parabéns!!!

Tomas Edison de

novembro 25 2021

[RESPONDER](#)

O simbolo # é para comentários ou para o comando sudo?

Carlos E.

novembro 26 2021

O símbolo "#" pode ser usado para comentários. O comando sudo não usa ele não, até onde eu sei 😊

Ismar Russano

março 27 2022

[RESPONDER](#)

Olá pessoal Buenos...!!! Seguinte: Eu sou músico e usuário de programas Linux, mais especificamente UbuntuStudio. Tem uma função no Jack Audio Connection, de script, que não entendo corretamente. Gostaria de uma explicação um pouco melhor, se possível! -Execute script after startup - /usr/bin/a2jmidid -e & -Execute script on Shutdown - kill a2jmidid (tem configuração que usa & ou não) O simbolo & e as vezes &&, pra mim não é claro e já me deu dor de cabeça...kkk!!! Obrigado, antecipadamente, pelas respostas e ajuda !!! Abraços

Carlos E.

abril 14 2022

Olá Ismar, tudo bem? Então, o símbolo && permite que você rode dois comandos ao mesmo tempo. Por exemplo, o código **apt update && apt upgrade** vai atualizar os aplicativos e fazer o upgrade para a versão mais recente. Já o operador & vai rodar o segundo comando em plano de fundo. Você pode ler mais sobre isso [neste link aqui](#). Pra fechar, se você não está usando nenhum comando depois de **/usr/bin/a2jmidid -e &**, ele vai significar a mesma coisa que **/usr/bin/a2jmidid -e!**

Ednei T Goncalves

abril 07 2023

[RESPONDER](#)

Bom dia Ariane. Te sigo nos artigos Tutoriais da Hostinger, e talvez possa me tirar uma dúvida que ainda não consegui solucionar, nem mesmo nas pesquisas pela web. Preciso centralizar as informações apresentada na tela do terminal do Linux (Putty), mas precisamente os dados apresentados pelo neofetch. Também estou tentando posicionar a tela minimizado do Putty no lado esquerdo do vídeo. Grato pela ajuda

Carlos E.

abril 14 2023

Olá, Ednei! para centralizar as informações apresentadas na tela do terminal Linux (Putty), você pode seguir estes passos: 1. Abra o cliente do PuTTY SSH e insira o IP SSH e a Porta SSH do seu servidor. 2. Na opção “Connection”, clique no sub menu ‘Data’. 3. No campo “Auto-login username”, digite seu nome de usuário administrador ou ‘root’. 4. No campo “Terminal-type string”, digite “xterm”. 5. Clique em “Open” para prosseguir.

maio 18 2023

[RESPOSTA](#)

BOM DIA MEU NOME E ANDRE ESTOU A ALGUM TEMPO PESQUISANDO SOBRE O LINUX POIS RECETEMENTE ADQUIRI UMA MAQUINA QUE É COMANDADA POR UM APLICATIVO DO LINUX CHAMADO DE LINUXCNC GOSTARIA DE TER MAIS INFORMAÇÕES SOBRE ESSA PARTE DO LINUX.

Bruno Santana
maio 19 2023

Oi, André! Nesse caso você pode conferir o site oficial do LinuxCNC, que conta com várias documentações, fóruns e suporte: <http://linuxcnc.org>. Boa sorte!

Deixe uma resposta

Comentário*

Nome*

Email*

☐ Ao inserir seu e-mail, você concorda em receber a newsletter da Hostinger. Você pode cancelar a inscrição a qualquer momento. Confira nossa [Política de Privacidade](#).

Enviar

Esse site utiliza o Akismet para reduzir spam. [Aprenda como seus dados de comentários são processados](#).

[Hospedagem de Sites Profissional](#)[Servidor VPS](#)[Host Minecraft](#)[VPS CyberPanel](#)[Hospedagem Cloud](#)[Cheap WordPress Hosting](#)[Email Profissional](#)[Hospedagem CMS](#)[Hospedagem de Loja Virtual](#)[Hospedagem Grátis](#)[Loja Virtual](#)[Criador de Sites](#)[Criador de Logo](#)[Gerador de Nomes para Empresas](#)

DOMÍNIOS

[Registro de Domínio](#)[Transferir Domínio](#)[Domínio Grátis](#)[Domínio .xyz](#)[Domínio Barato](#)[Extensões de Domínios](#)[WHOIS](#)[Certificado SSL Gratuito](#)

AJUDA

[Tutoriais](#)[Base de Conhecimento](#)[Denunciar Abuso](#)

INFORMAÇÕES

[Termos de Migração](#)[Migração de Site](#)[Status do Sistema](#)[Programa de Afiliados](#)[Programa de Parceria](#)[Formas de Pagamento](#)[Prêmio](#)[Opiniões de Clientes](#)[Preços](#)[Mapa do Site](#)

SOBRE NÓS

[Sobre a Hostinger](#)[Tecnologias Hostinger](#)[Carreiras](#)[Roadmap \(em inglês\).](#)[Fale Conosco](#)

[Política de Privacidade](#)

[Termos de Serviço](#)



© 2004-2023 [hostinger.com.br](#) - Hospedagem de Sites, Cloud e VPS premium e Serviços de Registro de Domínio.

Preços listados sem IVA