

| | |
|---------------------------------|-----------|
| O Que é Linux? | 2 |
| Como Usar o Linux? | 2 |
| Comandos Linux | 2 |
| 1. Comando pwd | 3 |
| 2. Comando cd | 3 |
| 3. Comando ls | 3 |
| 4. Comando cat | 4 |
| 5. Comando cp | 4 |
| 6. Comando mv | 4 |
| 7. Comando mkdir | 4 |
| 8. Comando rmdir | 5 |
| 9. Comando rm | 5 |
| 10. Comando touch | 5 |
| 11. Comando locate | 5 |
| 12. Comando find | 6 |
| 13. Comando grep | 6 |
| 14. Comando sudo | 6 |
| 15. Comando df | 6 |
| 16. Comando du | 6 |
| 17. Comando head | 7 |
| 18. Comando tail | 7 |
| 19. Comando diff | 7 |
| 20. Comando tar | 7 |
| 21. Comando chmod | 7 |
| 22. Comando chown | 8 |
| 23. Comando jobs | 8 |
| 24. Comando kill | 8 |
| 25. Comando ping | 9 |
| 26. Comando wget | 9 |
| 27. Comando uname | 9 |
| 28. Comando top | 9 |
| 29. Comando history | 9 |
| 30. Comando man | 10 |
| 31. Comando echo | 10 |
| 32. Comando zip, unzip | 10 |
| 33. Comando hostname | 10 |
| 34. Comando useradd, userdel | 10 |
| Dicas e truques de Bônus | 11 |
| Conclusão | 11 |

O Que é Linux?

O Linux é uma família inteira de sistemas operacionais Unix, que é baseado no Kernel do Linux. Isso inclui a maioria dos sistemas baseados em Linux, como Ubuntu, Fedora, Mint, Debian, entre outros. Mais precisamente, eles são chamados de distribuições ou simplesmente distros. Desde que foi lançado em 1991, ele continuou a ganhar popularidade devido a sua natureza em ser de código aberto. Por causa disso, as pessoas podem livremente modificá-lo e redistribuí-lo sob seu próprio nome.

Ainda que o Linux não supere o Windows em [participação de mercado](#), suas distribuições são extremamente populares entre os [provedores de hospedagem de sites](#). Graças à estabilidade, flexibilidade e segurança, [36.7% dos sites na internet](#) usam Linux como seus servidores de sistemas operacionais.

Como Usar o Linux?

Quando você usa um sistema operacional, você tem que mexer no **shell** – uma interface que dá ao usuário o acesso aos serviços do sistema operacional.

A maioria das distribuições do Linux usam uma Interface Gráfica de Usuário (GUI) como o seu shell, principalmente para fornecer uma facilidade de uso aos seus usuários.

Dito isso, ele é mais recomendado para usar uma linha de comando porque ele é mais poderoso e eficiente. As tarefas que exigem um processo de múltiplas etapas pela GUI podem ser feitas em questão de segundos digitando comandos pela linha de comando.

Então, se você está pensando em usar o Linux, aprender linhas de comandos básicos do Linux é uma das principais coisas que você deve saber. E é exatamente isso o que você confere abaixo.

Comandos Linux

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

Embora os passos abaixo possam ser 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**.

2. Comando cd

Para navegar pelo filesystem do 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 **Documents**. Para fazer isso, simplesmente digite **cd Photos**.

Outro cenário em que você quer mudar completamente de diretório, digamos, para **/home/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 mover um diretório acima
- Use **cd** para ir diretamente para a pasta home
- Use **cd-** (com um hífen) para mover para os diretórios anteriores.

Uma nota de aviso: o shell do Linux é sensível a tipos de caracteres. Por isso, você precisa digitar o nome do diretório exatamente como ele é escrito (usando letras minúsculas ou maiúsculas).

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 -al** vai listar todos os arquivos e diretórios com informações detalhadas como permissões, tamanho, proprietário, etc.

4. Comando cat

O **cat** (abreviação para concatenar) é um dos comandos Linux mais usados. Ele é usado para visualizar o conteúdo de um arquivo na saída padrão (stdout). Para executar esse comando, digite **cat** seguido pelo nome do arquivo e sua extensão. Por exemplo: **cat file.text**.

Aqui estão outras maneiras de usar o comando **cat**:

- **cat > filename** cria um novo arquivo
- **cat filename1 filename2>filename3** junta dois arquivos (1 e 2) e armazena a saída deles num novo arquivo (3)
- para converter um arquivo para usos de maiúscula ou minúscula, use **cat filename | tr a-z A-Z >output.txt**

5. Comando cp

Use o comando **cp** para copiar arquivo do diretório atual em que você estiver. Por exemplo, o comando **cp scenery.jpg /home/username/Pictures** vai criar uma cópia de **scenery.jpg** para o diretório **Pictures**.

6. Comando mv

O usuário habitual do comando **mv** é mover arquivos, ainda que ele possa também ser usado para renomear arquivos.

Os argumentos neste comando é similar ao comando **cp**. Você precisa digitar **mv**, o nome do arquivo e o diretório de destino. Por exemplo: **mv file.txt /home/username/Documents**.

Já para renomear arquivos, a sintaxe é **mv nomeantigo.ext nomenovo.ext**.

7. Comando mkdir

Use o comando **mkdir** para criar um novo diretório – como **mkdir Music**, que vai criar um novo diretório chamado **Music**.

Também existem comandos **mkdir** extras:

- Para gerar um novo diretório dentro de outro diretório, use este comando básico do Linux **mkdir Music/Newfile**
- use a opção **p** (parents) para criar um diretório entre dois diretórios existentes. Por exemplo, **mkdir -p Music/2020/Newfile** vai criar o novo arquivo “2020”.

8. Comando rmdir

Se você precisa apagar (deletar) um diretório, use o comando **rmdir**. Porém, o **rmdir** só permite que você apague diretórios vazios (sem conteúdos).

9. Comando rm

O comando **rm** é usado para apagar um diretório e todos os conteúdos que estiverem lá dentro. Se você só quer deletar o diretório – como uma alternativa ao **rmdir** – use **rm -r**.

Nota: Tenha muito cuidado com este comando e sempre verifique duas vezes em qual diretório você está. Isso irá deletar tudo e não há opção para desfazer.

10. Comando touch

O comando **touch** permite criar novos arquivos em branco através de uma linha de comando. Como exemplo, digite **touch /home/username/Documents/Web.html** para criar um arquivo **HTML** chamado **Web** dentro do diretório **Documents**.

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 caracteres, permitindo que você pesquise por um arquivo não sabendo 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:

- Para encontrar arquivos no diretório atual, use **find . -name notes.txt**
- Para procurar por diretórios, use **/ -type d -name notes. txt**

13. Comando grep

Outro dos principais comandos Linux e que são frequentemente usados pela sua utilidade. O comando **grep** permite que você procure através de todo texto dentro de um arquivo específico.

Para ilustrar, use **grep blue notepad.txt** para procurar pela palavra **blue** no arquivo **notepad**.

Linhas que contêm a palavra pesquisadas serão mostradas por completo.

14. Comando sudo

Correspondente a **SuperUser Do**, o comando **sudo** permite que que você execute tarefas que exigem permissões root ou administrativas. Porém, não é muito aconselhável usá-lo diariamente porque pode ser que um erro aconteça se você fizer algo de errado.

15. Comando df

Use o comando **df** para obter uma resposta da quantidade de espaço de disco usado no seu sistema. Esse valor será mostrado em KBs. Se você ver os resultados em MBs, digite **df -m**.

16. Comando du

Se você quer verificar o quanto de espaço um arquivo ou um diretório ocupa, o comando **du** é a resposta. Entretanto, o resumo do uso de disco vai mostrar números, ao invés do tamanho do

formato atual. Se você quer esse valor em bytes, KBs ou MBs, use o argumento **-h** na linha de comando.

17. Comando head

O comando **head** é usado para ver as primeiras linhas de um arquivo de texto. Por padrão, ele vai mostrar as primeiras 10 linhas, mas você pode mudar essa quantidade para qualquer outra. Por exemplo, se você quer ver apenas as 5 primeiras linhas, digite **head -n 5 nomedoarquivo.txt**.

18. Comando tail

O comando **tail** tem função similar ao comando **head**. Mas ele mostra nas últimas 10 linhas de um arquivo de texto. Por exemplo, **tail -n nomedoarquivo.txt**.

19. Comando diff

O comando **diff** (diferença) compara o conteúdo de dois arquivos linha por linha. Depois de analisar esses arquivos, ele vai mostrar as linhas que não são comuns entre eles. Os programadores frequentemente usam este comando quando precisam fazer pequenas alterações em programas. Assim, eles não precisam reescrever o código inteiro.

A forma mais simples deste comando é **diff arquivo1.ext arquivo2.ext**.

20. Comando tar

O comando **tar** é o comando mais usado para arquivar múltiplos arquivos em um **tarball** – um formato de arquivo Linux que é similar ao formato **zip**, mas a compressão é opcional.

Este comando é um tanto complexo. E tem uma longa lista de funções, como adicionar novos arquivos em um arquivo já existente, listar conteúdos em um arquivo, extrair conteúdos de um arquivo e muitos outros. Veja estes [exemplos práticos](#) para saber as outras funções dele.

21. Comando chmod

O comando **chmod** é outro comando essencial. Ele é usado para ler, escrever e executar permissões de arquivos e diretórios. Como esse comando é relativamente complicado, você pode ler o [tutorial completo](#) para saber como executá-lo corretamente.

22. Comando chown

No Linux, todos os arquivos são de propriedade de um usuário específico. O comando **chown** permite que você mude ou transfira a propriedade de um arquivo para um nome de usuário específico. Por exemplo, o comando **chown linuxuser2 file.ext** vai fazer com que o **linuxuser2** seja o proprietário do **file.ext**.

23. Comando jobs

O comando **jobs** vai mostrar todos os trabalhos junto com os seus status de desenvolvimento. Um job é basicamente um processo que é iniciado pelo shell.

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

Use o comando **ping** para verificar o status da conexão do seu servidor. Por exemplo, digitando **ping google.com**, o comando vai checar se você está ou não conectado ao Google e também medir o tempo de resposta.

26. Comando wget

A linha de comandos do Linux é muito útil: você até mesmo pode baixar arquivos da internet com a ajuda do comando **wget**. Para fazer isso, simplesmente digite **wget** seguido pelo link de download do arquivo.

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.

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. É muito útil para monitorar o uso de espaço no sistema, especialmente para saber qual processo deve ser encerrado porque ele consome muitos recursos.

29. Comando history

Quando você já estiver familiarizado com o Linux, vai perceber que você pode executar centenas de comandos todos os dias. Por exemplo, o comando **history** (histórico) é particularmente útil se você quer rever quais comandos já usou antes.

30. Comando man

Está confuso com as funções de cada comando mostrado? Sem problema. Você pode aprender como usar o comando certo usando o comando **man**. Por exemplo, digitar **man tail** vai mostrar um manual de instruções do comando **tail**.

31. Comando echo

Este comando é usado para mover alguns dados para um arquivo. Por exemplo, se você quiser adicionar o texto “Olá, meu nome é John” em um arquivo chamado name.txt, digite **echo Olá, meu nome é John >> name.txt**

32. Comando zip, unzip

Use o comando **zip** para compactar seus arquivos em um arquivo zip e use o comando **unzip** para extrair arquivos descompactados de um arquivo zip.

33. Comando hostname

Se você quiser saber o nome do seu host/network (da sua rede), basta digitar **hostname**. E se adicionar **-I** ao final, exibirá o endereço IP da sua rede.

34. Comando useradd, userdel

Como o Linux é um sistema multiusuário, isso significa que mais de uma pessoa pode interagir com o mesmo sistema ao mesmo tempo. **useradd** é usado para criar um novo usuário, enquanto **passwd** adiciona uma nova senha à conta deste usuário. Para criar um novo usuário chamado João, adicionar **user** e adicione seu tipo de senha, **passwd 12345678**.

Remover um usuário é muito semelhante a adicionar um novo usuário. Para excluir o tipo de conta de usuário, **userdel username**.

Dicas e truques de Bônus

Use o comando **clear** para limpar o terminal se estiver cheio de muitos comandos usados anteriormente.

Experimente o botão **TAB** para preencher automaticamente o que você está digitando. Por exemplo, se você precisa digitar Documentos, comece a digitar o comando (vamos usar o **cd Docu**, então aperte a tecla **TAB**) e o terminal preencherá o restante, mostrando o **cd Documents**.

Ctrl + C e **Ctrl + Z** são usados para qualquer comando que esteja funcionando no momento. Ctrl + C interromperá o comando com segurança, e o Ctrl + Z forçará a parada.

Se você congelar seu terminal acidentalmente com o **Ctrl + S**, simplesmente desfaça o congelamento com **Ctrl + Z**.

Ctrl + A move você para o início da linha enquanto **Ctrl + E** move você para o fim.

Você pode executar vários comandos em um único comando usando o “;” para separá-los. Por exemplo, **command1, command2, command3**. Ou use **&&** se desejar que o próximo comando seja executado apenas quando o primeiro comando indicado estiver funcionando.

Conclusão

Os Comandos Linux ajudam os usuário a executar qualquer tarefa do sistema operacional de forma prática, fácil e com muita eficiência. Pode levar um tempo para aprender os comandos básicos, mas nada é impossível para quem pratica com frequência.

No fim das contas, conhecer e aperfeiçoar estes comandos acima vai trazer uma série de benefícios a você como usuário. Então, não desista e continue usando para memorizar os comandos Linux, suas funções e resultados.

Até mais!