



Imagens de Yosi Azwan / Getty

Antes de entrarmos nisso, quero deixar claro: usar a linha de comando não é um requisito para usar o Linux. Menciono isso porque a ideia de digitar comandos tende a assustar novos usuários e meu objetivo - durante anos - tem sido apresentar às pessoas o sistema operacional de código aberto.

Dito isto, existem certos comandos do Linux com os quais posso contar para usar todos os dias. Esses comandos me ajudam a fazer o que faço, a manter meus sistemas funcionando bem e a garantir que estou informado sobre o que está acontecendo em minhas máquinas.

Além disso: Os primeiros 5 comandos do Linux que todo novo usuário deve aprender

É claro que a experiência computacional de cada pessoa é diferente, então o que você precisa provavelmente será diferente dos comandos dos quais dependo. Dito isso, aqui estão os seis comandos do Linux que são quase garantidos para serem executados diariamente no meu teclado.

O que você vai ler:



- [1. topo](#)
- [2. ssh](#)
- [3.sudo](#)
- [4. apto](#)
- [5. wget](#)
- [6. sistemactl](#)

1. topo

Sempre gosto de saber o que está acontecendo nos bastidores, especialmente se sinto que algo deu errado. Quando essa hora chegar, *principal* é meu comando de escolha. Com *principal* posso descobrir rapidamente quantos recursos do sistema um aplicativo ou comando está usando. Além disso, posso ver o PID (ID do processo) associado a esse comando ou aplicativo e posso usá-lo para encerrar o aplicativo, caso seja necessário.

A razão pela qual eu uso *principal* em vez de um dos [aplicativos](#) GUI, é que posso acessar remotamente uma máquina e descobrir as informações do terminal. Top é rápido, fácil de usar e nunca me falha.

Para abrir o topo, emita o seguinte comando:

```
principal
```

2. ssh

Eu poderia argumentar *ssh* sendo o comando mais importante da lista. Por que? Considere o seguinte: às vezes o VirtualBox enlouquece e uma VM convidada começa a se comportar mal o suficiente para travar meu sistema. Quando isso acontecer, posso usar *ssh* para acessar o sistema, use *principal* para descobrir o PID associado à VM e, em seguida, elimine a VM com o *matar PID* (onde PID é o ID do processo do convidado do VirtualBox) – e está tudo pronto.

Muitas vezes tenho que me conectar remotamente a outras máquinas na minha [rede](#) (ou fora da minha rede) para realizar certas tarefas (como atualizar um servidor). Eu também uso *scp* (que faz parte *ssh*) para mover arquivos na minha rede, então, sim, *ssh* é muito importante.

SSH é simples de usar. Por exemplo, se eu quiser acessar remotamente um servidor na minha LAN, posso emitir o seguinte:



```
ssh jack@192.168.1.100
```

3. sudo

Este é provavelmente o comando que executo com mais frequência - porque estou sempre instalando [software](#), atualizando aplicativos, gerenciando processos e serviços e fazendo todo tipo de coisa que requer privilégios de administrador. Se não fosse por *sudoeu* teria que primeiro mudar para o usuário root, o que pode ser um problema de segurança. Com *sudoganho* privilégios de administrador temporários, posso executar um comando ou aplicativo e saber que esses privilégios serão revogados automaticamente após um período definido. Sudo foi uma adição muito inteligente ao Linux e continua a ser um dos comandos mais importantes que executei até hoje.

Além disso: 5 comandos relacionados à rede Linux que todo novo usuário deve saber

Sudo é simples de usar. Você simplesmente adiciona-o ao início de qualquer comando que você executaria e que requer privilégios de administrador, como este:

```
sudo apt instalar atualização -y
```

4. apto

Dada a frequência com que testo e reviso software de código aberto em distribuições baseadas em Debian/Ubuntu, não deveria ser surpresa que *apto* é um dos meus comandos mais usados. O *apto* gerenciador de pacotes simplifica o processo de gerenciamento de aplicativos e até mesmo de correção de instalações quebradas (*sudo apt instalar -f* salvou minha pele em diversas ocasiões). Embora as interfaces GUI do apt sejam excelentes, há algumas coisas que elas não podem fazer (como *apt purgar apt autoremove*), e é por isso que geralmente prefiro gerenciar pacotes na linha de comando.

Apto é fácil. Por exemplo, se quiser instalar o GIMP, você emitiria o seguinte comando:

```
sudo apt instalar gimp -y
```

5. wget

Wget é um daqueles comandos que podem não ser usados todos os dias, mas quando é usado, percebo o quanto é importante. Quando há um arquivo ou script que preciso baixar e



não há um link acessível pelo navegador da Web para usar, *wget* pode fazer o trabalho. eu uso *wget* regularmente, especialmente ao instalar software baseado em servidor, onde o sistema operacional provavelmente não possui uma GUI da qual depender. Com *wget* não importa se existe um ambiente de desktop ou não; Ainda posso pegar tudo o que preciso para fazer o trabalho.

Digamos que você queira baixar o código-fonte da versão mais recente do GIMP. Você pode fazer isso com:

```
wget
```

6. *systemctl*

Systemctl me permite controlar processos. Não só posso iniciá-los e pará-los, mas também verificar se estão funcionando ou não. Se um processo morreu, o *systemctl* irá reportá-lo. Então, também posso usá-lo para ajudar a solucionar o motivo pelo qual o processo foi interrompido (com *journctl* - que faz parte *systemctl*) definir o destino padrão (como GUI ou terminal), mascarar ou desmascarar um serviço, ativar um serviço para iniciar na inicialização, listar arquivos de unidade e muito mais.

Além disso: Meus 5 editores de texto Linux favoritos (e por que você deveria usar um)

Systemctl é obrigatório para manter um sistema funcionando bem. Aqueles que são novos no Linux provavelmente não precisarão se preocupar com *systemctl* no início, mas eventualmente você vai querer se aprofundar nos detalhes desse comando porque ele é muito útil.

O comando *systemctl* é simples. Digamos que você queira iniciar o daemon SSH. Isso pode ser feito com:

```
sudo systemctl iniciar ssh
```