



Certified Tech Developer

The Ultimate Degree

Infraestrutura I

Objetivos

Nos exercícios a seguir, vamos nos familiarizar um pouco mais com os comandos mais usados em um terminal Linux. Veremos também como gerar um espaço para consultas. Além de listar e gerenciar serviços Linux.

Banco de ajuda para lembrar alguns dos comandos mais usados do Linux

1. Comando pwd

Para encontrar o caminho do diretório de trabalho atual (pasta) onde você está. O comando retornará um caminho absoluto (completo), que é basicamente um caminho para todos os diretórios que começam com uma barra (/). Um exemplo de caminho absoluto é **/home/username**.

2. Comando cd

Para navegar pelos arquivos e diretórios do Linux. Ele solicitará o caminho completo ou o nome do diretório, dependendo do diretório de trabalho atual em que você está.



Digamos que você esteja em **/home/username/Documents** e queira ir para **Photos**, um subdiretório de **Documents**. Para isso, basta digitar o seguinte comando: **cd Photos**.

Outro cenário é se você quiser ir para um diretório completamente novo, por exemplo, **/home/username/Filmes**. Neste caso, você deve escrever **cd** seguido do caminho absoluto do diretório: **cd / home / username / Movies**.

Existem alguns atalhos para ajudá-lo a navegar rapidamente:

- **cd..** (com dois pontos) para subir um diretório.
- **cd** para ir diretamente para a pasta pessoal.
- **cd -** (com um traço) para ir para o diretório anterior.

Como uma nota de rodapé, o shell do Linux diferencia maiúsculas de minúsculas. Portanto, você deve digitar o nome do diretório exatamente.

3. Comando ls

Para visualizar o conteúdo de um diretório. Por padrão, este comando exibirá o conteúdo do seu diretório de trabalho atual.

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

Existem variações que você pode usar com o comando:

- **ls -R** também listará todos os arquivos em subdiretórios.
- **ls -a** mostrará arquivos ocultos.
- **ls -al** listará os arquivos e diretórios com informações detalhadas, como permissões, tamanho, proprietário, etc.

4. Comando cat

Este é um dos comandos mais usados no Linux. Seu nome vem da abreviação do termo em inglês: *concatenate*. Usado para listar o conteúdo de um arquivo na saída padrão (stdout). Para executar este comando, digite **cat** seguido do nome do arquivo e sua extensão. Por exemplo: **cat file.txt**.

Aqui estão outras maneiras de usar o comando:

- **cat > filename** cria um novo arquivo.
- **cat filename1 filename2 > filename3** junta dois arquivos (1 e 2) e armazena sua saída em um novo arquivo (3)
- Para converter um arquivo para maiúsculas ou minúsculas, escrevemos: **cat filename | tr az AZ > output.txt**

5. Comando cp

Para copiar arquivos do diretório atual para um diretório diferente. Por exemplo, o comando **cp cenário.jpg / home / nome de usuário / Imagens** criaria uma cópia do **cenário.jpg** (do seu diretório atual) no diretório **Imagens**.

6. Comando mv

Seu principal uso é para mover arquivos, embora também possa ser usado para renomear arquivos. Os argumentos em mv são semelhantes ao comando cp. Você deve escrever mv, o nome do arquivo e o diretório de destino. Por exemplo: **arquivo mv.txt / home / nome de usuário / Documentos**.

Para renomear os arquivos, o comando do Linux é **mv oldname.ext newname.ext**



7. Comando mkdir

Para criar um novo diretório: se você digitar **mkdir Music**, ele criará um diretório chamado **Music**.

Existem também comandos adicionais de **mkdir**:

- 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** criará a nova pasta **2020**.

8. Comando rmdir

Se você precisar remover um diretório, use o comando **rmdir**. No entanto, só permite remover diretórios vazios.

9. rm comando

O comando **rm** é usado para diretórios remover e conteúdos dentro deles. Se você deseja apenas remover o diretório, como alternativa ao rmdir, use **rm -r**.

Nota: Tenha muito cuidado com este comando e verifique em qual diretório você está. Este comando remove tudo e não pode ser desfeito.

10. comando touch

Permite que você crie um novo arquivo em branco através da linha de comando do Linux. Como exemplo, **digite touch /home/username/Documents/Web.html** para criar um arquivo HTML intitulado **Web** no diretório Documentos.

11. comando locate

Você pode usar este comando para **localizar** um arquivo, assim como o comando `search` no Windows. Além disso, usar o argumento `-i` em conjunto com este comando fará com que ele não faça distinção entre maiúsculas e minúsculas, para que você possa pesquisar um arquivo mesmo que não consiga lembrar seu nome exato.

Para localizar um arquivo que contenha duas ou mais palavras, use um asterisco (*). Por exemplo, o comando **`locate -i school * note`** localizará qualquer arquivo que contenha a palavra “school” e “note”, em maiúsculas ou minúsculas.

12. comando find

Semelhante ao comando anterior, também permite pesquisar arquivos e diretórios. A diferença é que o comando `find` é usado para procurar arquivos dentro de um determinado diretório. Por exemplo, o comando **`find / home / -name notes.txt`** procurará um arquivo chamado `notes.txt` dentro do diretório inicial e seus subdiretórios.

Outras variações do uso de `find` são:

- Para procurar arquivos no diretório atual: **`find. -name notes.txt`**
- Para pesquisar diretórios: **`/ -type d -name notes.txt`**

13. Comando grep

Permite pesquisar todo o texto em um determinado arquivo. Para ser claro: **`grep blue notepad.txt`** procurará a palavra `blue` no arquivo do bloco de notas. As linhas contendo a palavra pesquisada serão exibidas.

14. sudo

Abreviação **SuperUser Do** (fazer como super usuário). Este comando permite que você execute tarefas que requerem permissões administrativas ou de root. No entanto, não é aconselhável usar este comando para uso diário, pois pode ser fácil ocorrer um erro se você fizer algo errado.

15. Comando df

Use o comando df para obter um relatório sobre o uso do espaço em disco do sistema. Isso mostrará o relatório em porcentagem e KB. Se você quiser ver o relatório em megabytes, digite df **-m**.

16. Comando du

Permite verificar quanto espaço um arquivo ou diretório ocupa. No entanto, o resumo de uso do disco mostrará os números dos blocos de disco em vez do formato de tamanho normal. Se você quiser vê-lo em bytes, kilobytes e megabytes, adicione o argumento **-h** à linha de comando.

17. comando head

Usado para visualizar as primeiras linhas de qualquer arquivo de texto. Por padrão, ele mostrará as primeiras dez linhas, mas você pode alterar esse número ao seu gosto. Por exemplo, se você deseja exibir apenas as primeiras cinco linhas, **digite head -n 5 filename.ext.**



18. comando tail

Tem uma função similar ao comando head, mas ao invés de mostrar as primeiras linhas, mostrará as últimas dez linhas de um arquivo de texto. Por exemplo, **tail -n nomedoarquivo.ext**.

19. Comando diff

Comparar o conteúdo de dois arquivos linha por linha. Depois de analisar os arquivos, ele gera as linhas incompatíveis. Os programadores costumam usar esse comando quando precisam fazer modificações no programa em vez de reescrever todo o código-fonte. A maneira mais simples de usar este comando é **diff arquivo1.ext arquivo2.ext**

20. comando tar

Este comando é o mais comumente usado para salvar vários arquivos em um **tarball**, um formato de arquivo Linux semelhante ao formato ZIP, com compactação opcional. Este comando é bastante complexo com uma longa lista de funções, como adicionar novos arquivos a um arquivo existente, listar o conteúdo de um arquivo, extrair o conteúdo de um arquivo e muito mais.

21. Comando chmod

É usado para alterar as permissões de leitura, escrita e execução de arquivos e diretórios. Como este comando é bastante complicado, você pode encontrar mais informações para executá-lo corretamente.

22. Comando chown

No Linux, todos os arquivos pertencem a um usuário específico. Este comando permite alterar ou transferir a propriedade de um arquivo para o nome de usuário especificado. Por exemplo, **chown linuxuser2 file.ext** tornará **linuxuser2** o proprietário do **arquivo.ext**.

23.jobs

O comando **jobs** exibe todos os jobs atuais junto com seus estados. Um trabalho é basicamente um processo iniciado pelo shell.

24. Comando kill

Se você tem um programa que não está respondendo, você pode fechá-lo manualmente usando o comando **kill**. Isso enviará um certo sinal para o programa que está funcionando mal e instruirá o aplicativo a ser encerrado.

Há um total de sessenta e quatro sinais que você pode usar, mas as pessoas geralmente usam apenas dois sinais:

- **SIGTERM** (15): solicita que um programa pare de ser executado e lhe dá algum tempo para salvar todo o seu progresso. Se você não especificar o sinal ao digitar o comando kill, este sinal será usado.
- **SIGKILL** (9): força os programas a parar imediatamente. O progresso não salvo e será perdido.

Além de conhecer os sinais, você também deve saber o número de identificação do processo (PID) do programa que deseja parar. Se você não conhece o PID, basta executar o comando **ps ux**. Após saber qual sinal deseja utilizar e o PID do programa, digite a seguinte sintaxe: **kill [opção de sinal] PID**.

25. Comando ping

Use o comando ping para verificar seu status de conectividade com um servidor. Por exemplo, simplesmente digitando ping google.com, o comando verificará se você pode se conectar ao Google e também medirá o tempo de resposta.

26. Comando wget

Permite baixar arquivos da Internet. Para fazer isso, basta digitar **wget** seguido do link de download.

27. Comando uname

Ele imprimirá informações detalhadas sobre seu sistema Linux, como nome da máquina, sistema operacional, kernel, etc. Seu nome vem da abreviação de " Unix name".

28. Comando top

Este é um terminal equivalente ao Gerenciador de Tarefas no Windows. O comando top listará os processos em execução e a quantidade de CPU que cada processo usa. É muito útil monitorar o uso dos recursos do sistema, principalmente para saber qual processo deve ser finalizado, pois consome muitos recursos.

29. Comando history

Ao usar o Linux por um certo período de tempo, você notará rapidamente que pode executar centenas de comandos todos os dias. Este comando permite revisar os comandos inseridos anteriormente.

30. Comando man

Confuso sobre a função de certos comandos do Linux? Não se preocupe, você pode aprender facilmente como usá-los diretamente do shell do Linux usando este comando. Por exemplo, digitar **man tail** exibirá instruções manuais para o comando **tail**.

31. Comando echo

Este comando é usado para mover alguns dados para um arquivo. Por exemplo, se você quiser adicionar o texto "Hello, my name is David" em um arquivo chamado **name.txt**, você deve digitar **echo Hello, my name is David >> name.txt**

32. Comando zip, unzip

Use este comando para compactar seus arquivos em um arquivo ZIP e use o comando **unzip** para extrair os arquivos compactados de um arquivo ZIP.

33. Comando hostname

Se você quiser saber o nome do seu **host/rede**, basta digitar **hostname**. Adicionar um **-I** no final mostrará 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 senha à conta desse usuário.

Por exemplo, para adicionar uma nova pessoa chamada David, **digitei useradd David**. Então, para adicionar a senha dele, **digitei passwd 123456789**.



A exclusão de um usuário é muito semelhante à adição de um novo usuário. Para excluir a conta de usuário, escreva **userdel David**.

35. Dicas e truques extras

- Use o comando **clear** para limpar o terminal se ele estiver cheio de muitos comandos passados.
- Experimente o botão **TAB** para completar automaticamente o que você está digitando. Por exemplo, se você precisar digitar "Documents", comece a digitar um comando - por exemplo "cd Doc" -, pressione a tecla **TAB** e o terminal completará o resto, mostrando **cd Documents**.
- **Ctrl + C** e **Ctrl + Z** são usados para interromper qualquer comando que esteja sendo executado no momento. **Ctrl + C** interrompe o comando com segurança, enquanto **Ctrl + Z** o forçará a parar.
- Se acidentalmente congelar seu terminal usando **Ctrl+S**; apenas desfaça este comando com **Ctrl + Q**.
- **Ctrl + A** move você para o início da linha, enquanto **Ctrl + E** move você para o final.
- Você pode executar vários comandos em um único comando usando ponto e vírgula(;) para separá-los. Por exemplo: **Comando1; Comando2; Comando 3**. Ou use **&&** se você quiser que o próximo comando seja executado apenas quando o primeiro for bem-sucedido.

Instruções

Individualmente, abra seu terminal Linux em sua máquina virtual ou WSL. Em `pwd / home / user1 /`

Exercício 1

Listar serviços no Linux

Mude para usuário root:

```
su root
```

Agora você pode listar todos os serviços no Linux. Para fazer isso, execute o comando:

```
sudo systemctl list-unit-files --type service --all
```

Ao executar o comando, você verá todos os serviços que estão no sistema. No entanto, você também verá que alguns têm um estado definido. Vamos ver o que tudo isso significa:

- Serviços habilitados (Enabled) são aqueles que estão em execução no momento. Estes geralmente não têm problemas.
- Serviços desabilitados (Disabled) são aqueles que não estão ativos, mas podem ser ativados a qualquer momento sem nenhum problema.
- Os serviços mascarados (Masked) não são executados, a menos que retirem essa propriedade.
- Os serviços estáticos (Static) só serão utilizados se for necessário outro serviço ou unidade.



- Finalmente, existem serviços gerados (generated) através de um script de inicialização SysV ou LSB com o gerador systemd.

Caso você queira apenas saber quais serviços estão ativos, você deve usar um comando junto com o grep, como este:

```
sudo systemctl | grep running
```

Exercício 2

Gerenciando serviços do Linux

Agora que você sabe como listar serviços no Linux, é hora de aprender como gerenciar um serviço específico. Tenha em mente que cada serviço representa um software e cada software funciona de forma diferente. Neste tutorial mostraremos apenas como iniciar, verificar o status e parar os serviços, ou seja, os controles básicos.

Para iniciar um serviço no Linux, você precisa executar o seguinte comando:

```
sudo systemctl start [service_name]
```

Se o serviço estiver configurado corretamente, ele será iniciado. Agora, se você quiser pará-lo, use o seguinte comando:

```
sudo systemctl stop [service_name]
```

Por outro lado, para verificar o status de um serviço você pode usar:

```
sudo systemctl status [service_name]
```



Também é possível executar um serviço enquanto o sistema operacional está carregando:

```
sudo systemctl enable [service_name]
```

Ou remova-o do carregamento inicial:

```
sudo systemctl disable [service_name]
```

Finalmente, é possível monitorar uma porta que está sendo usada por um serviço. Para isso, use tcpdump. Para instalá-lo no Ubuntu, basta executar:

```
sudo apt-get install tcpdump
```

Se você estiver usando o CentOS 7, execute:

```
sudo yum install tcpdump
```

Em seguida, execute o seguinte comando:

```
sudo tcpdump -d (para ver a lista de interfaces de rede  
disponíveis)  
sudo tcpdump -i enp0s3 porta 80
```

A saída fornecerá todas as informações de rede necessárias.



Conclusão

Os comandos básicos do Linux ajudam os usuários a realizar tarefas com facilidade e eficácia. Pode demorar um pouco para lembrar alguns dos comandos básicos, mas nada é impossível com muita prática. No final, conhecer e dominar esses comandos, sem dúvida, será benéfico para sua experiência.

Aprender a listar serviços no Linux é fácil e isso pode acelerar bastante a solução de problemas. Agora você pode gerenciar seu VPS Linux como um profissional.