



Os shells do [Linux](#) podem parecer semelhantes ao prompt de comando, mas se você vier da linha de comando do Windows, há muito o que descobrir sobre eles.

O que você vai ler:



- [1 Linux tem vários shells](#)
- [2 Delimitadores de caminho são diferentes](#)
- [3 Shells Linux são desenvolvidos ativamente](#)
- [4 Você pode executar o PowerShell a partir de Linux e shells Linux no Windows](#)
- [5 Shells Linux têm controle de trabalho](#)
- [6 Linux tem comandos internos e externos](#)
- [7 Você tem uma escolha em linguagens de script](#)

1

Linux tem vários shells

```
david@debian:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
/usr/bin/tmux
/usr/bin/pwsh
/opt/microsoft/powershell/7/pwsh
```



Uma diferença do prompt de comando do Windows no Linux é que você pode escolher entre vários shells. Embora seja possível fazer isso no Windows desde a época do MS-DOS, a capacidade de alternar shells no Linux é muito mais proeminente.

Embora Bash seja o shell padrão mais popular para distribuições Linux, existem outros shells que você pode usar, como tcsh, zsh e fish, entre outros. Todos eles têm seus próprios recursos. Muitos deles podem emular o shell Bourne para compatibilidade com scripts de shell necessários para tarefas administrativas. Você pode escolher um shell diferente com o comando chsh.

2

Delimitadores de caminho são diferentes

Se você observar os nomes de caminho no Windows e no Linux, notará que os delimitadores são diferentes. No Windows, os caminhos são separados por caracteres “\” (barra invertida), enquanto no Linux e outros sistemas semelhantes ao Unix, eles são separados por um caractere “/” (barra ou apenas “barra”). Isso ocorre principalmente por razões históricas.

Se você está acostumado com o Windows, isso pode te atrapalhar. Os comandos do Linux também diferenciam maiúsculas de minúsculas, o que significa que caracteres minúsculos e maiúsculos têm significados diferentes.

3

Shells Linux são desenvolvidos ativamente



Embora a Microsoft tenha descontinuado amplamente o antigo prompt de comando em favor do PowerShell, os shells do Linux ainda estão ativamente desenvolvidos. É por isso que você pode escolher entre tantos shells no Linux hoje.

A linha de comando não é uma relíquia histórica, mas um importante [software](#) usado diariamente por usuários avançados, administradores e desenvolvedores. Bugs estão sendo descobertos e corrigidos, e novos recursos são adicionados constantemente.

4

Você pode executar o PowerShell a partir de Linux e shells Linux no Windows

Você não precisa apenas executar o PowerShell no Windows ou shells do Linux no Linux. Você pode executar um do outro.

```
PS /> dir

Directory: /

UnixMode          User Group      LastWriteTime         Size Name
-----
lrwxrwxrwx        root root      10/7/2024 10:12           7 bin -> usr/bin
drwxr-xr-x        root root      1/13/2025 11:17          4096 boot
drwxr-xr-x        root root      1/13/2025 11:18          3380 dev
drwxr-xr-x        root root      1/13/2025 11:28         12288 etc
drwxr-xr-x        root root      10/7/2024 11:00          4096 home
lrwxrwxrwx        root root      10/7/2024 10:12           7 lib -> usr/lib
lrwxrwxrwx        root root      10/7/2024 10:12           9 lib64 -> usr/lib64
drwx-----      root root      10/7/2024 10:12        16384 lost+found
drwxr-xr-x        root root      10/7/2024 10:12          4096 media
drwxr-xr-x        root root      10/7/2024 10:12          4096 mnt
drwxr-xr-x        root root      1/13/2025 11:28          4096 opt
dr-xr-xr-x        root root      1/13/2025 11:18           0 proc
drwx-----      root root      12/18/2024 15:23          4096 root
drwxr-xr-x        root root      1/13/2025 11:18           720 run
lrwxrwxrwx        root root      10/7/2024 10:12           8 sbin -> usr/sbin
drwxr-xr-x        root root      10/7/2024 10:12          4096 srv
dr-xr-xr-x        root root      1/13/2025 11:18           0 sys
drwxrwxrwt        root root      1/13/2025 11:28          4096 tmp
drwxr-xr-x        root root      10/7/2024 10:12          4096 usr
drwxr-xr-x        root root      10/7/2024 10:12          4096 var
lrwxrwxrwx        root root      1/13/2025 11:09          30 initrd.img ->
boot/initrd.img-6.1.0-29-amd64
lrwxrwxrwx        root root      1/13/2025 11:09          30 initrd.img.old ->
boot/initrd.img-6.1.0-28-amd64
lrwxrwxrwx        root root      1/13/2025 11:09          27 vmlinuz ->
boot/vmlinuz-6.1.0-29-amd64
lrwxrwxrwx        root root      1/13/2025 11:09          27 vmlinuz.old ->
boot/vmlinuz-6.1.0-28-amd64

PS /> |
```

A Microsoft mantém uma versão do PowerShell para Linux e você pode instalá-lo como um shell na maioria das principais distribuições Linux. Você pode instalá-lo usando os repositórios de pacotes da Microsoft para sua [distribuição](#).

Você também pode executar shells do Linux no Windows. A maneira mais fácil de fazer isso é instalar o Windows Subsystem para Linux e, em seguida, instalar uma distribuição Linux. Conforme mencionado anteriormente, a maioria das distros usa Bash por padrão, mas você pode alterar o shell para o que quiser.

Você pode até instalar a versão Linux do PowerShell no Windows se quiser tornar as coisas ainda mais estranhas. Assim, você pode executar a versão Linux do Powershell no Linux no Windows.

Com WSL, você pode explorar o Linux a partir do Windows e combinar os dois.

5

Shells Linux têm controle de trabalho



```
david@debian:~$ sleep 100
^Z
[1]+  Stopped                  sleep 100
david@debian:~$ bg
[1]+  sleep 100 &
david@debian:~$
```

Uma coisa que separa os shells do Linux do prompt de comando é que eles foram criados para multitarefa desde o início. O resultado disso é que os shells modernos possuem um recurso chamado controle de trabalho que permite colocar uma tarefa em segundo plano enquanto você executa outros comandos.

Você pode iniciar um comando em segundo plano anexando à linha no shell um caractere & (e comercial). Por exemplo,

```
some_long_operation &
```

O shell executará esse comando em segundo plano e retornará um número de trabalho entre colchetes. Você pode usar isso para informar ao shell o que fazer com o trabalho.

Para trazer a parte de trás para o primeiro plano, você pode usar o comando `fg`. Por padrão, `fg` trará de volta o trabalho que você acabou de colocar em segundo plano, mas se você tiver vários trabalhos, precisará especificar um número de trabalho prefixado por % (sinal de porcentagem). Por exemplo, para o trabalho 1 em primeiro plano, digite:

```
fg %1
```

Para colocar um trabalho atualmente em execução em segundo plano, pressione `Ctrl+Z` e digite “`bg`” no shell.

6

Linux tem comandos internos e externos



memória
virtual

```
david@debian:~$ which ls
/usr/bin/ls
david@debian:~$
```

Os comandos de controle de trabalho que mostrei foram integrados à maioria dos shells modernos. Eles são chamados de “builtins” por esse motivo. Eles devem funcionar em Bash e zsh.

A razão pela qual os desenvolvedores de shell criam comandos internos é que geralmente é mais rápido executá-los do que comandos externos, uma vez que o shell já está na memória e não precisa ser carregado. Alguns dos comandos internos são duplicados em programas convencionais do Linux, e isso pode ser fácil confundir os dois. A melhor opção é ler a documentação do seu shell e aprender quais comandos internos ele possui.

Freqüentemente, você também pode executar um comando chamado `which` que informa que tipo de comando é. Por exemplo, para BG:

```
which bg
```

O shell reportará que é um comando interno. No zsh, que também é um comando interno:

```
which which
```

7

Você tem uma escolha em linguagens de script

Se você já teve que automatizar algo no prompt de comando, pode ter escrito um arquivo em lote ou um `.bat` com a extensão do arquivo.

Você também pode automatizar tarefas com scripts de shell. Os shells do Linux são poderosos e possuem muitos recursos de programação completos, como arrays e hashes/matrizes associativas. E se precisar de mais capacidade, você pode aprender linguagens de script como Perl ou Python. Haverá muito para mantê-lo ocupado enquanto você explora a linha de comando do Linux, não importa o idioma que você usa.