

# Introdução ao terminal Linux

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree

# Índice

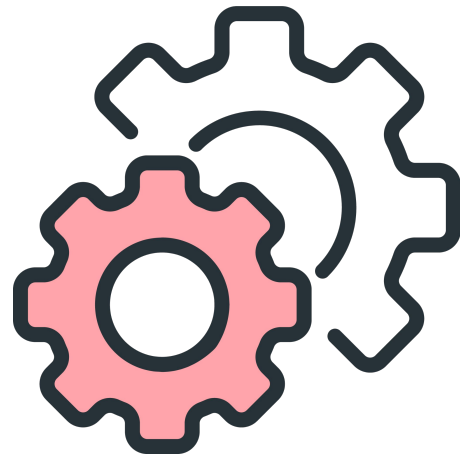
1. [Console](#)
2. [Diferentes tipos de shell](#)
3. [Executando o console](#)
4. [Elevando privilégios](#)

# 1 | Console

# Console Linux

A interface de linha de comando, ou CLI – por sua sigla em inglês `command-line interface` – é um método de comunicação entre usuário e máquina que aceita instruções do usuário por meio de linhas de texto (seguindo certas regras de sintaxe que podem ser interpretadas pelo operador sistema).

A ferramenta que habilita a função de interface do usuário é chamada de shell. Aplicado no escopo da interface de linha de comando, estaríamos falando de um shell CLI ou interpretador de comandos.



## 2 | Diferentes tipos de shell



# Bourne Shell

É nomeado após seu criador no Bell Labs, Steve Bourne. Foi o primeiro shell usado para o sistema operacional Unix e superou em grande parte muitos dos shells mais recentes. Todas as versões do Linux Unix permitem que os usuários mudem para o Bourne Shell original, conhecido simplesmente como "sh", se assim o desejar. No entanto, lembre-se de que isso renunciará a recursos, como conclusão de nome de arquivo e histórico de comandos, adicionados por repositórios subsequentes.

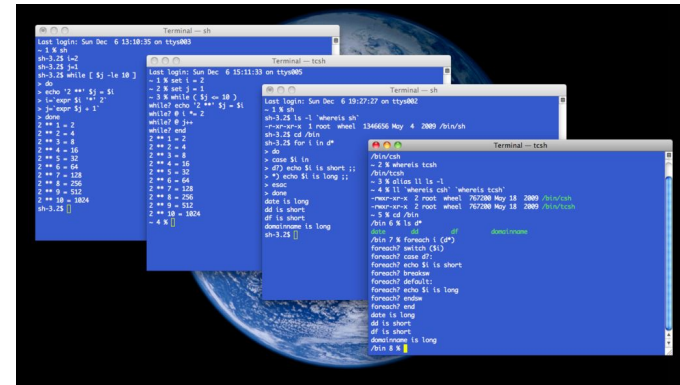


```
Terminal
-rwxr-xr-x 1 bin 18296 Jun 8 1979 fsck
-rwxr-xr-x 1 bin 1458 Jun 8 1979 getty
-rw-r--r-- 1 root 49 Jun 8 1979 group
-rwxr-xr-x 1 bin 2482 Jun 8 1979 init
-rwxr-xr-x 1 bin 8484 Jun 8 1979 mkfs
-rwxr-xr-x 1 bin 3642 Jun 8 1979 mknod
-rwxr-xr-x 1 bin 3976 Jun 8 1979 mount
-rw-r--r-- 1 root 141 Jun 8 1979 passwd
-rw-r--r-- 1 bin 366 Jun 8 1979 rc
-rw-r--r-- 1 bin 266 Jun 8 1979 ttys
-rwxr-xr-x 1 bin 3794 Jun 8 1979 umount
-rwxr-xr-x 1 bin 634 Jun 8 1979 update
-rw-r--r-- 1 bin 40 Sep 22 05:49 utmp
-rwxr-xr-x 1 root 4520 Jun 8 1979 wall
# ls -l /*unix*
-rwxr-xr-x 1 sys 53302 Jun 8 1979 /hptunix
-rwxr-xr-x 1 sys 52850 Jun 8 1979 /hptmunix
-rwxr-xr-x 1 root 50990 Jun 8 1979 /rkunix
-rwxr-xr-x 1 root 51982 Jun 8 1979 /rl2unix
-rwxr-xr-x 1 sys 51790 Jun 8 1979 /rphtunix
-rwxr-xr-x 1 sys 51274 Jun 8 1979 /rptmunix
# ls -l /bin/sh
-rwxr-xr-x 1 bin 17310 Jun 8 1979 /bin/sh
#
```

# C/TC Shell

O C Shell foi desenvolvido após o Bourne Shell e tem como objetivo facilitar o controle do sistema para o programador da linguagem C. A razão disso é que sua sintaxe, como veremos, é muito semelhante à dessa linguagem.

Também conhecido popularmente como csh, está presente em outros sistemas operacionais, por exemplo, no Mac OS. Possui uma evolução, conhecida como tcsh, que incorpora funcionalidades avançadas e maiores atalhos de teclado.





# Korn Shell

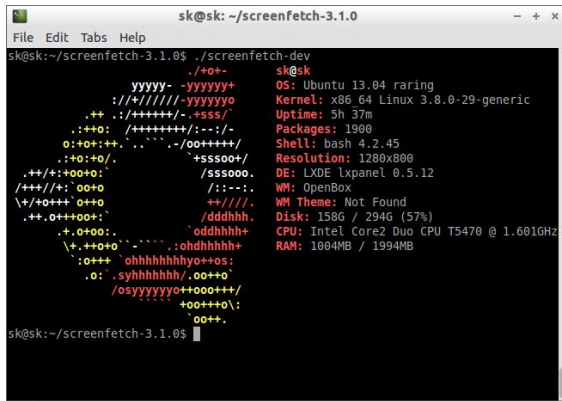
Também foi escrito pelo programador da Bell Labs, David Korn. Tente combinar os recursos do C Shell, TC Shell e Bourne Shell em um único pacote. Ele também incluiu a capacidade de criar novos comandos de shell para unwrappers quando necessário.

```
QEMU
# uname -a
OpenBSD berkeley.my.domain 5.3 GENERIC#53 amd64
# pwd
/tmp
# cd ~
# ls -l
total 2112
-rw-r--r-- 1 root wheel 112 Jun 20 2013 .Xauthority
-rw-r--r-- 1 root wheel 22 Mar 13 2013 .Xdefaults
-rw-r--r-- 1 root wheel 578 Mar 13 2013 .cshrc
drwxr-xr-x 3 root wheel 512 Jun 24 2013 .gem
-rw-r--r-- 1 root wheel 125 Mar 13 2013 .login
-rw-r--r-- 1 root wheel 328 Mar 13 2013 .login
-rw-r--r-- 1 root wheel 552 Jun 24 2013 .profile
-rw-r--r-- 1 root wheel 63 Jun 20 2013 .serverauth.18088
-rw-r--r-- 1 root wheel 1044692 Jun 28 2013 mbox
# whoami
root
# ls -l $(which vi)
-r-xr-xr-x 3 root bin 361496 Mar 13 2013 /usr/bin/vi
# echo $SHELL
/bin/ksh
#
```



Possui funções avançadas para lidar com arquivos de comandos ou combiná-los com linguagens de programação especializadas, como AWK e Perl.

# Bourne-Again Shell (BASH)



```
sk@sk: ~/screenfetch-3.1.0
File Edit Tabs Help
sk@sk:~/screenfetch-3.1.0$ ./screenfetch-dev
      .+o+~
    yyyyy- -yyyyyy+
    ://////////-yyyyyyo
    .++ ://+++++/-.-+sss/
    .:++o: /++++++/-:--:/-
    o:++o:++' .-./oo+++++/
    .:++o:++/ .+sssoo++/
    .++/+++oo+o: /sssoooo.
    /+++//+ 0o+o /+/-+/-
    \+/o++ 0+o+o +++++/
    .++o++o++' /ddhddd.
    .+o+o+: .oddhddd.
    \+..+o+o' .:ohdhhhhh+
    .o+++`ohhhhhhhhyo++os:
    .o: .syhhhhhhh/.oo+o+
    /osyyyyyyo++ooo+++/
    .+++++
    +oo++o\
    oo++.
```

O Bourne-Again Shell é uma versão atualizada do Bourne Shell original. É um shell amplamente utilizado na comunidade de código aberto.

Sua sintaxe é semelhante à utilizada pelo Bourne Shell, incorporando funcionalidades mais avançadas encontradas nos shells C, TC e Korn.



A funcionalidade adicional que faltava ao Bourne incluía a capacidade de completar os nomes dos arquivos pressionando a tecla TAB, a capacidade de lembrar um histórico de comandos recentes e executar vários programas em segundo plano ao mesmo tempo.

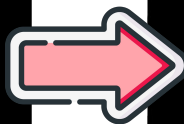
# 3 | Executando o console

# Console Linux: Executar na inicialização

Embora cada distribuição Linux tenha sua própria maneira de acessar o console, quando o SO inicia nos níveis 1, 2, 3 e 4, ele nos levará ao console por padrão.

```
CentOS Linux (3.10.0-1160.6.1.el7.x86_64) ? (Core)
CentOS Linux (0-rescue-ff31bca5aa6fe84a9c27749f5242b30b) ? (Core)

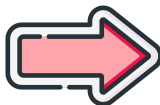
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```



```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.6.1.el7.x86_64 on an x86_64
localhost login: _
```

# Console Linux: Executar a partir da GUI

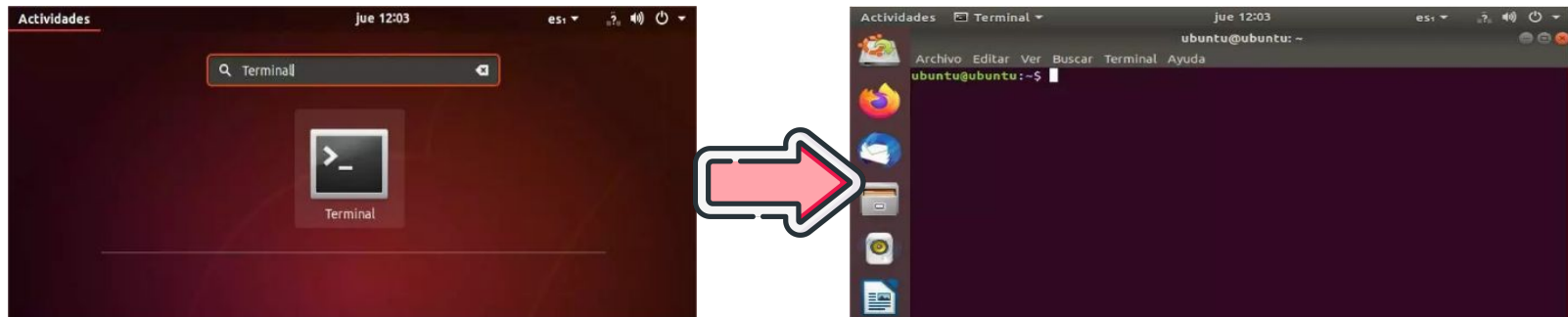
- Se, em vez disso, nosso sistema operacional iniciar no nível 5 (com GUI), para usar o terminal, temos diferentes opções. Estes variam de acordo com a distribuição instalada. No caso do Ubuntu, temos duas opções:
- A primeira é lançar um TTY, ou espaço de trabalho sem ambiente gráfico. Podemos executar 7 terminais ao mesmo tempo dessa maneira. De 1 a 6, nenhum tem interface gráfica. Para alterar o TTY no Linux devemos usar o atalho de teclado Control + Alt mais a tecla –de F1 a F7– do TTY que queremos executar.



```
ubuntu 18.04.4 LTS ubuntu tty3
ubuntu login: _
```

# Console Linux: Executar a partir da GUI

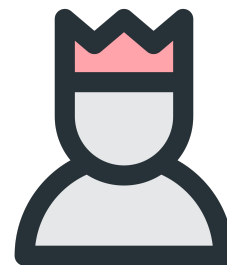
- A segunda opção é encontrar um aplicativo dedicado que seja executado em uma janela, dentro do painel de aplicativos da nossa distribuição. No caso do Ubuntu, por exemplo, podemos encontrar este terminal na gaveta de programas do ambiente gráfico GNOME.



# 4 | Elevando privilegios

# Os privilégios do superusuário root

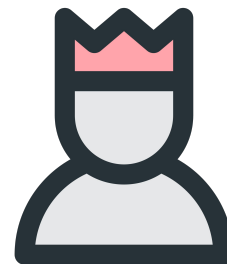
Em geral, os sistemas operacionais contemplam o uso de apenas um usuário, que possui permissões de administrador. No Linux, as coisas são tratadas de uma maneira particular, a conta de usuário comum é separada da conta de superusuário e isso é o que conhecemos como root. Esta conta tem todos os privilégios e permissões para realizar ações no sistema.





# Os privilégios do superusuário root

Para a execução de alguns comandos, devemos inserir o referido acesso (senha de root). No entanto, é necessário ter conhecimento sobre as ações que são realizadas, pois uma ação realizada de forma errada pode causar danos significativos ao sistema. O uso de instruções com privilégios de superusuário pode ser extremamente útil, mas totalmente devastador se não soubermos as consequências de seu uso no sistema. Vamos ver o método de elevar nossos privilégios.



# Elevando privilégios

Supondo que façamos login como um usuário “comum”, chamado “edorio” e desejamos reiniciar um serviço (cron), obteremos o seguinte:



{}

```
edorio@DESKTOP-W10:~$ service cron start
* Starting periodic command scheduler cron
cron: can't open or create /var/run/crond.pid: Permission denied
[fail]
edorio@DESKTOP-W10:~$
```

# Elevando privilégios

Para evitar o erro, devemos usar o comando sudo, antes do comando que queremos executar. Ele nos pedirá a senha de root e será executado com sucesso.



{ }

```
edorio@DESKTOP-W10:~$ sudo service cron start
[sudo] password for edorio:
 * Starting periodic command scheduler cron
[ OK ]
edorio@DESKTOP-W10:~$
```

DigitalHouse>