

# Capítulo 2. Acesso a linha de comando

## Resumo

Meta	Fazer login em um sistema Linux e executar comando simples no shell
Objetivos	<ul style="list-style-type: none"><li>• Fazer login em um sistema Linux e executar comando simples no shell</li><li>• Fazer login no sistema Linux com o ambiente GNOME de área de trabalho e executar comandos a partir de um prompt do shell em um programa do terminal</li><li>• Economizar tempo ao executar comandos a partir de um prompt do shell com atalhos Bash</li></ul>
Seções	<ul style="list-style-type: none"><li>• Acesso à linha de comando (e teste)</li><li>• Acesso a linha de comando com a area de trabalho (e exercicio orientado)</li><li>• Execução de comandos com o shell Bash (e teste)</li></ul>
Laboratorio	Acesso a linha de comando

## Introdução ao shell Bash

Uma *linha de comando* é uma interface baseada em texto que pode ser usada para inserir instruções em um sistema de computador.

- O shell exibe uma string quando está aguardando a entrada do usuário, chamada de *prompt do shell*. Quando um usuário regular inicia um shell, o prompt inclui um caractere de dólar final ( `$` ):

```
[user@host ~]$
```

- Um caractere hash ( `#` ) substitui o caractere de dólar ( `$` ) se o shell for executado pelo superusuário, `root`. Esse caractere indica que é um shell de superusuário, o que ajuda a evitar erros que pode afetar todo o sistema.

```
[root@host ~]#
```

## Bash

- O shell `bash` fornece uma linguagem de script que pode auxiliar a automação de tarefas.
- O shell tem recursos que podem simplificar ou possibilitar operações difíceis de realizar em escala com ferramentas gráficas.
- `bash` tem uma linguagem de script sofisticada e é mais semelhante ao Windows PowerShell
- A partir do macOS 10.15 Catalina, a Apple alterou o shell padrão para o shell `zsh`, um shell alternativo que também está disponível no RHEL.

## Noções básicas do shell:

Os comandos inseridos no prompt do shell têm três partes básicas:

- Comando a executar
- Opção para ajudar o comportamento do comando
- Argumentos

O comando é o nome do programa a ser executado. Ele pode ser seguido de uma ou mais opções, que ajustam o comportamento do comando ou o que ele faz. As opções geralmente começam com um ou dois traços (`-a` ou `--all`, por exemplo) para distingui-las dos argumentos.

Os comandos também podem ser seguidos por um ou mais argumentos, que com frequência indicam um alvo no qual o comando deverá operar.

**Ex:** `usermod -L user01`, `usermod` é o comando, `-L` é a opção e `user01` é o argumento. Esse comando bloqueia a senha da conta de usuário `user01`

## Faça login em um sistema local:

- Um *terminal* é uma interface baseada em texto
- Usado para inserir comandos em um sistema de computador e imprimir saídas dele.
- Para executar o shell, você precisa fazer login no computador em um terminal.

Um teclado de hardware e monitor para entrada e saída pode estar diretamente conectados a ele (Esse é o *console físico* da máquina do Linux).

## O que é o console físico:

- oferece suporte a vários *consoles virtuais*, que podem ser executados em terminais separados.
- Cada console virtual é compatível com uma sessão de login independente podendo ser alternada pressionando **Ctrl+Alt** e uma tecla de função (**F1 a F6**) ao mesmo tempo
- O computador pode fornecer um prompt de login gráfico em um dos consoles virtuais. Você pode usar o prompt de login gráfico para fazer login em um *ambiente gráfico*.
- **O ambiente gráfico também é executado em um console virtual.** Para obter um prompt do shell, você deve iniciar um programa de terminal no ambiente gráfico. O prompt do shell é fornecido em uma janela de aplicativo do seu programa de terminal gráfico.
  - Muitos administradores de sistemas optam por não executar um ambiente gráfico em seus servidores, porque os usuários não fazem login em servidores como um espaço de trabalho de área de trabalho. A carga de trabalho de um servidor pode usar com mais eficiência os recursos significativos que um ambiente gráfico usa.

## Login em um sistema remoto:

No Linux, a maneira mais comum de obter um prompt do shell em um sistema remoto é usar o Secure Shell (SSH).

Nesse exemplo, um usuário com um prompt do shell na máquina `host` usa `ssh` para fazer login no sistema Linux remoto `remotehost` com o usuário `remoteuser`:

```
[user@host ~]$ssh remoteuser@remotehost
remoteuser@remotehost's password:password
[remoteuser@remotehost ~]$
```

- O comando `ssh` criptografa a conexão para proteger a comunicação contra espionagem ou sequestro de senhas e conteúdo.

**Uma maneira alternativa de autenticar em uma máquina remota sem digitar uma senha é por meio de *autenticação de chave***

## ***pública.***

- Com esse método de autenticação, os usuários têm um arquivo de identidade especial com uma *chave privada*, que é equivalente a uma senha, e que eles mantêm em segredo.
- Sua conta no servidor é configurada com uma *chave pública* correspondente, o que não precisa ser secreto.
- Ao fazer login, os usuários podem configurar `ssh` para fornecer a chave privada. Se a chave pública correspondente estiver instalada nessa conta nesse servidor remoto, ela será registrada sem a solicitação de uma senha.

## **Exemplo de autenticação com chave publica:**

- A opção `-i` do comando `ssh` é usada para especificar o arquivo de chave privada do usuário, que é `mylab.pem`. A chave pública correspondente já está configurada como uma chave autorizada na conta `remoteuser`.

```
[user@host ~]$ssh -i mylab.pem remoteuser@remotehost
[remoteuser@remotehost ~]$
```

- Para que a conexão funcione, somente o usuário proprietário do arquivo pode ter acesso para ler o arquivo de chave privada. No exemplo anterior, em que a chave privada está no arquivo `mylab.pem`, o comando `chmod 600 mylab.pem` pode ser usado para garantir que somente o proprietário possa ler o arquivo
- Toda vez que você se conecta a um host remoto com `ssh`, o host remoto envia sua *chave de host* para se autenticar e ajudar a configurar a comunicação criptografada
- O comando `ssh` compara a chave de host a uma lista de chaves de host salvas para garantir que ele não tenha sido alterado.
  - Se a chave do host tiver sido alterada, isso pode indicar que alguém está tentando fingir ser esse host para sequestrar a conexão que também é conhecida como ataque interceptor.

# **Modulo 02 - Parte2**

## **Acesso à linha de comando com a área de trabalho**

Fazer login no sistema Linux com o ambiente GNOME de área de trabalho e executar comandos a partir de um prompt do shell em um programa do terminal.

## Introdução ao ambiente GNOME de área de trabalho

O *ambiente de desktop* é a interface gráfica do usuário em um sistema Linux.

### Acessar espaços de trabalho

- Os *espaços de trabalho* são telas separadas que têm janelas de aplicativo diferentes. Eles podem ser usados para organizar seu ambiente de trabalho agrupando janelas de aplicativo abertas por tarefas.
- Por exemplo, você pode agrupar janelas para uma determinada atividade de manutenção do sistema (como a configuração de um novo servidor remoto) em um espaço de trabalho, enquanto e-mails e outros aplicativos de comunicação podem ser agrupados em outro espaço de trabalho.
- Escolha entre dois métodos para alternar entre espaços de trabalho:
  - O primeiro método é pressionar **Ctrl+Alt+LeftArrow** ou **Ctrl+Alt+RightArrow** para alternar entre espaços de trabalho em sequência.
  - O segundo é alternar para a visão geral **Activities** desejada e clicar no espaço de trabalho escolhido.
  - Uma vantagem de usar a visão geral **Activities** é que as janelas podem ser clicadas e arrastadas entre o espaço de trabalho usando o **workspace selector** à direita da tela, e a **windows overview**, no centro da tela.

### Inicialização de um terminal:

Para obter um prompt do shell do GNOME, inicie um aplicativo de terminal gráfico, como o **GNOME Terminal**. Use um dos seguintes métodos para iniciar um terminal:

- Na visão geral **Activities**, selecione `Terminal` em **dash**, em Favorites ou com o botão **Show Applications**.
- Pesquise por `terminal` no campo de pesquisa na parte superior de **windows overview**.
- Pressione a combinação de teclas **Alt+F2** para abrir **Enter a Command** e digite `gnome-terminal`.
- O prompt do shell e a barra de título da janela de terminal indicam o nome de usuário atual, o nome do host e o diretório de trabalho.

## Bloqueio da tela e logout:

- Bloqueie a tela, ou faça o logout totalmente, no menu de sistema no lado direito da barra superior.
- Para bloquear a tela, no menu do sistema no canto superior direito, clique no botão de bloqueio na parte inferior do menu ou pressione **Super+L** (que pode ser mais fácil de lembrar como **Windows+L**). A tela também é bloqueada se a sessão gráfica ficar inativa por alguns minutos
- Uma **lock screen curtain** (cortina de tela de bloqueio) aparece, mostrando a hora do sistema e o nome do usuário conectado. Para desbloquear a tela, você pode pressionar **Enter**, **Space** ou clicar com o botão esquerdo do mouse. Em seguida, digite a senha desse usuário na **lock screen**.
- Para fazer o logout e finalizar a sessão de login gráfico atual, selecione o menu do sistema no canto superior direito da barra superior e clique em **Power Off/Log Out** → **Log Out**. Uma janela é exibida, oferecendo a opção de **Cancel** ou confirmar a ação **Log Out**.

## Desligamento ou reinicialização do sistema

- Para desligar o sistema, no menu do sistema no canto superior direito, selecione **Power Off/Log out** → **Power Off** ou pressione **Ctrl+Alt+Del**. Uma janela é exibida, oferecendo a opção de **Cancel** ou confirmar a ação **Power Off**. Se você não fizer uma escolha, o sistema será desligado automaticamente após 60 segundos.

Para reiniciar o sistema, no menu do sistema no canto superior direito, selecione **Power Off/Log out → Restart**. Uma janela é exibida, oferecendo a opção de **Cancel** ou confirmar a ação **Restart**. Se você não fizer uma escolha, o sistema será reiniciado automaticamente após 60 segundos.

## Exercício orientado: Acesso à linha de comando com a área de trabalho

Neste exercício, você se conecta pelo gerenciador de tela gráfico como um usuário normal para se familiarizar com o ambiente de área de trabalho GNOME Standard, fornecido pelo GNOME 40.

Com o usuário `student` na máquina `workstation`, use o comando `lab` para preparar seu sistema para este exercício.

Esse comando prepara seu ambiente e garante que todos os recursos necessários estejam disponíveis.

```
[student@workstation ~]$ lab start cli-desktop
```

## Instruções:

1. Altere a senha de `student` de `student` para `55TurnK3y`.

```
### **Importante**
```

```
0 script `finish` redefine a senha do usuário `student` para
```

1.1 Abra uma janela do `Terminal` e use o comando `passwd` no prompt do shell.

1.2 Na janela `Terminal` exibida, digite `passwd` no prompt do shell. Altere a senha de `student` de `student` para `55TurnK3y`.

2. Faça o logout e faça login novamente como `student` com `55TurnK3y` como a senha para verificar a senha alterada.

- 2.1 Clique no menu do sistema no canto superior direito.
  - 2.2 Selecione **Power Off/Log Out** → **Log Out**.
  - 2.3 Clique em **Log Out** na caixa de diálogo de confirmação exibida.
  - 2.4 Na tela de login do GNOME, clique na conta de usuário `student`. Digite `55TurnK3y` quando a senha for solicitada.
  - 2.5 Pressione ENTER
- 
3. Bloqueie a tela.
    - 3.1 No menu do sistema no canto superior direito, pressione o botão **Lock**.
- 
4. Desbloqueie a tela.
    - 4.1 Pressione **Enter** para desbloquear a tela.
    - 4.2 No campo **Password**, digite a senha `55TurnK3y`.
    - 4.3 Pressione **Enter**.
- 
5. Determine como desligar a `workstation` na interface gráfica, mas pressione **Cancel** para cancelar a operação sem desligar o sistema.
    - 5.1 No menu do sistema no canto superior direito, selecione **Power Off/Log Out** → **Power Off**. Uma caixa de diálogo é exibida com as opções de **Cancel** (Reiniciar) ou **Power Off** (Desligar) a máquina.
    - 5.2 Clique em **Cancel** na caixa de diálogo exibida.

## Encerramento

Na máquina `workstation`, altere para o diretório pessoal do usuário `student` e use o comando `lab` para concluir este exercício. Essa etapa é importante para garantir que recursos de exercícios anteriores não afetem exercícios futuros.

## Execução de comandos com o shell Bash

### Objetivos

Economizar tempo ao executar comandos a partir de um prompt do shell com atalhos Bash.



## Sintaxe básica de comandos

- O `GNU Bourne-Again Shell` ( `bash` ) é um programa que interpreta comandos digitados pelos tipos usuário.
- Cada string digitada no shell pode ter até três partes:
  - Comando
  - As opções (que geralmente começam com um caractere de hífen `-` ou hífen duplos `--` )
  - Argumento
- Cada palavra digitada no shell é separada das outras palavras por espaços.
- Comandos = são os nomes dos programas que estão instalados no sistema. Cada comando tem suas opções e argumentos.

### Para digitar mais comandos em uma linha:

- use um ponto e vírgula (;) como separador de comandos.
- Um ponto e vírgula é um membro de uma classe de caracteres chamada **metacaracteres** que tem uma interpretação especial para o `bash`.

## Escrita de comandos simples:

O comando `date` mostra a data e a hora atuais. O superusuário ou um usuário com privilégios também pode usar o comando `date` para definir o relógio do sistema. Use um sinal de mais ( `+` ) como um argumento para especificar uma string de formatação para o comando `date`.

```
[user@host ~]$ date
Sun Feb 27 08:32:42 PM EST 2022
[user@host ~]$ date +%R
20:33
[user@host ~]$ date +%x
02/27/2022
```

O comando `passwd` sem opções altera a senha do usuário atual. Para alterar a senha, primeiro especifique a senha original da conta. Por padrão, o

comando `passwd` é configurado para solicitar uma senha segura, composta por letras minúsculas, maiúsculas, números e símbolos e que não seja baseada em uma palavra do dicionário. Um superusuário ou usuário com privilégios pode usar o comando `passwd` para alterar a senha de outro usuário.

```
[user@host ~]$passwd
Changing password for useruser.
Current password:old_password
New password:new_password
Retype new password:new_password
passwd: all authentication tokens updated successfully.
```

O comando `file` verifica o cabeçalho compilado de um arquivo para um `magic number` de 2 dígitos e exibe seu tipo. Os arquivos de texto são reconhecidos porque não são compilados.

```
[user@host ~]$file /etc/passwd
/etc/passwd: ASCII text
[user@host ~]$file /bin/passwd
/bin/passwd: setuid ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked, interpreter /lib64/ld
-linux-x86-64.so.2, BuildID[sha1]=a467cb9c8fa7306d41b96a820
b0178f3a9c66055, for GNU/Linux 3.2.0, stripped
[user@host ~]$file /home
/home: directory
```

## Visualização do conteúdo dos arquivos

O comando `cat` é frequentemente usado no Linux. Use esse comando para criar múltiplos arquivos ou arquivos únicos, visualizar o conteúdo dos arquivos, concatenar o conteúdo de vários arquivos e redirecionar o conteúdo do arquivo a um terminal ou a arquivos.

```
[user@host ~]$cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...output omitted...
```

## Para exibir o conteúdo de vários arquivos:

Adicione os nomes dos arquivos ao comando `cat` como argumentos:

```
[user@host ~]$cat file1 file2cat
Hello World!!
Introduction to Linux commands.
```

O comando `less` exibe uma página de um arquivo de cada vez e permite que você percorra as páginas. Use o comando `less` para avançar e voltar nas páginas de arquivos mais longos que cabem em uma janela de terminal. Use as teclas de **UpArrow** e **DownArrow** para rolar para cima e para baixo. Pressione **q** para sair do comando.

Os comandos

`head` e `tail` exibem o início e o fim de um arquivo, respectivamente. Por padrão, esses comandos exibem 10 linhas do arquivo, mas ambos têm uma opção `-n` que permite especificar um número diferente de linhas.

```
[user@host ~]$head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
[user@host ~]$tail -n 3 /etc/passwd
gdm:x:42:42:./var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:980:978:./run/gnome-initial-setup:/s
```

```
bin/nologin
dnsmasq:x:979:977:Dnsmasq DHCP and DNS server:/var/lib/dnsm
asq:/sbin/nologin
```

O comando `wc` conta linhas, palavras e caracteres em um arquivo.  
Use as opções:

- `-l` = exibir apenas as linhas
- `-w` = exibir apenas as palavras
- `-c` = exibir apenas caracteres determinados

```
[user@host ~]$wc /etc/passwd
41   98 2338 /etc/passwd
[user@host ~]$wc -l /etc/passwd ; wc -l /etc/group
41 /etc/passwd
63 /etc/group
[user@host ~]$wc -c /etc/group /etc/hosts
883 /etc/group
114 /etc/hosts
997 total
```

## Noções básicas de preenchimento com Tab:

Com o preenchimento com Tab, os usuários podem completar rapidamente comandos ou nomes de arquivo depois de digitar uma parte suficiente no prompt para torná-los exclusivos. Se os caracteres digitados não forem únicos, pressionar a tecla **Tab** duas vezes mostra todos os comandos iniciados pelos caracteres digitados.

O preenchimento com Tab ajuda a

```
[user@host ~]$ pasTab+Tab1
passwd      paste      pasuspender
[user@host ~]$ passTab2
[user@host ~]$ passwd
Changing password for user user.
Current password:
```

**1** Pressione **Tab** duas vezes.

**2** Pressione **Tab** uma vez.

completar os nomes de arquivo ao digitá-los como argumentos de comandos. Pressione

**Tab** para completar o máximo do nome do arquivo quanto possível. Pressionar **Tab** uma segunda vez fará com que o shell liste todos os arquivos que correspondam ao padrão atual. Digite caracteres adicionais até o nome ser exclusivo e use o preenchimento com Tab para completar o comando.

```
[user@host ~]$ ls /etc/pasTab 1
[user@host ~]$ ls /etc/passwdTab 2
passwd passwd-
```

1 Pressione **Tab** uma vez.

2 Pressione **Tab** uma vez.

Use o comando `useradd` para criar usuários no sistema. O comando `useradd` tem muitas opções que podem ser difíceis de lembrar. Usando o preenchimento com Tab, você pode concluir o nome da opção com o mínimo de digitação.

```
[root@host ~]#useradd --Tab+Tab
--badnames          --gid                      --no-log-in
it                  --shell
--base-dir          --groups                    --non-unique
e                   --skel
--btrfs-subvolume-home --help                --no-user-group
roup               --system
--comment           --home-dir            --password
--uid
--create-home       --inactive              --prefix
--user-group
--defaults          --key                      --root
--expiredate        --no-create-home         --selinux-user
ser
```

## Escrita de um comando longo em várias linhas

Comandos com muitas opções e argumentos podem rapidamente se tornar longos e são automaticamente envolvidos pela janela de comando quando o cursor atinge a margem direita. Em vez disso, digite um comando longo usando mais de uma linha para facilitar a leitura.

- Para gravar um comando em mais de uma linha, use um caractere de barra invertida (`\`), que é chamado de *caractere de escape*. O caractere de barra invertida ignora o significado do caractere a seguir.

```
[user@host ~]$head -n 3 \usr/share/dict/words \usr/share/
dict/linux.words
==> /usr/share/dict/words <==
1080
10-point
10th

==> /usr/share/dict/linux.words <==
1080
10-point
10th
```

## Exibição do histórico de comandos:

O comando `history` exibe uma lista de comandos executados anteriormente precedidos por um número de comando.

O caractere ponto de exclamação (!) é um metacaractere usado para expandir comandos anteriores sem precisar redigitá-los:

- O comando `! number` expande o comando que corresponde ao número indicado.
- O comando `! string` expande o comando mais recente que começa com a string especificada.

```
[user@host ~]$history
...output omitted...
 23  clear
 24  who
 25  pwd
```

```

26  ls /etc
27  uptime
28  ls -l
29  date
30  history
[user@host ~]$!ls
ls -l
total 0
drwxr-xr-x. 2 student student 6 Feb 27 19:24 Desktop
...output omitted...
[user@host ~]$!26
ls /etc
abrt                                hosts                                pulse
adjtime                            hosts.allow                        purple
aliases                            hosts.deny                        qemu-ga
...output omitted...

```

As teclas de seta ajudam a navegar pelos comandos anteriores no histórico do shell:

- A **UpArrow** edita o comando anterior na lista do histórico
- A **DownArrow** edita o próximo comando na lista do histórico
- As teclas **LeftArrow** e **RightArrow** movem o cursor para a esquerda e para a direita no comando atual da lista de histórico para que você possa editar o comando antes de executá-lo

## Use a combinação de teclas:

- **Esc+.** ou **Alt+.** simultaneamente para inserir a última palavra do comando anterior na localização atual do cursor. O uso repetido da combinação de teclas substituirá o texto pela última palavra dos comandos anteriores do histórico.
- A combinação de teclas **Alt+.** é particularmente conveniente, porque você pode segurar **Alt** e pressionar repetidamente para percorrer facilmente pelos comandos anteriores.

## Edição da linha de comando:

Quando usado de forma interativa, `bash` tem um recurso de edição de linha de comando. Use os comandos de editor de texto para se mover e modificar o comando atualmente digitado.

O movimento no comando atual e a passagem pelo histórico de comandos usando as teclas de seta foram introduzidos anteriormente nesta seção. A tabela a seguir mostra outros comandos de edição poderosos.

**Tabela 2.1. Atalhos úteis de edição de linha de comando**

Atalho	Descrição
Ctrl + A	Ir para o início da linha de comando
Ctrl + E	Ir para o final da linha de comando
Ctrl + U	Limpar do cursor ao início da linha de comando
Ctrl + K	Limpar do cursor ao final da linha de comando
Ctrl + LeftArrow	Ir para o início da palavra anterior na linha de comando
Ctrl + RightArrow	Ir para o final da palavra anterior na linha de comando
Ctrl + R	Pesquisar um padrão na lista de historico de comandos

## Sumário

- O shell Bash é um interpretador de comandos que solicita aos usuários interativos que especifiquem os comandos do Linux.
- Muitos comandos têm uma opção `-help` que exibe uma tela ou mensagem de uso.
- Você pode usar espaços de trabalho para organizar várias janelas de aplicativos.
- O botão **Activities** no canto superior esquerdo da barra superior fornece um modo de visão geral que ajuda a organizar janelas e iniciar aplicativos.
- O comando `file` varre o início de um arquivo e exibe seu tipo.
- Os comandos `head` e `tail` exibem o início e o fim de um arquivo, respectivamente.
- Você pode usar o preenchimento com Tab para preencher nomes de arquivos ao digitá-los como argumentos para os comandos.



- Você pode usar a interface gráfica para muitas tarefas administrativas. Você pode desativar a interface para preservar recursos para aplicativos em execução.
- Você pode escrever muitos comandos na mesma linha usando o caractere ponto-e-vírgula ; e executar um único comando em várias linhas usando o caractere de barra invertida \ .