

**重要**

上一屏幕示例显示了如何向典型用户显示延续的命令。不过，在学习材料（例如本教程）中描述这种现实情况通常会引起混淆。在键入命令时，新学员可能会误插入额外的大于号字符。shell会将键入的大于号字符解读为进程重定向，而这并不是用户的本意。我们会在下一章讨论进程重定向。

为了避免造成这种混淆，本教程不会在屏幕输出中显示辅助提示符。用户仍会在其 shell 窗口中看到辅助提示符，但教材故意只显示键入的字符，如下例所示。请与之前的屏幕示例进行比较。

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

命令历史记录

history 命令显示之前执行的命令的列表，带有命令编号作为前缀。

感叹号字符 (!) 是元字符，用于扩展之前的命令而不必重新键入它们。**!number** 命令扩展至与指定编号匹配的命令。**!string** 命令扩展至最近一个以指定字符串开头的命令。

```
[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 user user 6 Mar 29 21:16 Desktop
...output omitted...
[user@host ~]$ !26
ls /etc
abrt          hosts          pulse
```

```
adjtime          hosts.allow      purple
aliases          hosts.deny       qemu-ga
...output omitted...
```

编辑历史记录列表中的上一个命令。编辑历史记录列表中的下一个命令。在历史列表中的当前命令中左右移动光标，以便您在运行命令之前进行编辑。

您可以使用 **Esc+.** 或 **Alt+.** 组合键，在光标的当前位置插入上一命令的最后一个单词。重复使用组合键可将该文本替换为历史记录中更早命令的最后一个单词。**Alt+.** 组合键尤其方便，因为您可以按住 **Alt** 键，再反复按 **.** 键来轻松地回滚到更早的命令。

编辑命令行

以交互方式使用时，**bash** 具有命令行编辑功能。这允许用户使用文本编辑器命令在当前键入的命令内移动并进行修改。使用方向键可以在当前命令内移动，也可访问命令历史记录，这在本课程的前面部分已作介绍。下表介绍了更为强大的编辑命令。

命令行编辑实用快捷键

快捷键	描述
Ctrl+A	跳到命令行的开头。
Ctrl+E	跳到命令行的末尾。
Ctrl+U	将光标处到命令行开头的内容清除。
Ctrl+K	将光标处到命令行末尾的内容清除。
Ctrl+← 向左箭头	跳到命令行中前一字的开头。
Ctrl+→ 向右箭头	跳到命令行中下一字的末尾。
Ctrl+R	在历史记录列表中搜索某一模式的命令。

还有其他几个命令行编辑命令可用，但这些是对新用户最有用的命令。其他命令可以在 **bash(1)** man page 中找到。



参考文献

bash(1)、**date(1)**、**file(1)**、**cat(1)**、**more(1)**、**less(1)**、**head(1)**、**passwd(1)**、**tail(1)** 和 **wc(1)** man page

► 小测验

使用 BASH SHELL 执行命令

选择以下问题的正确答案：

► 1. 哪一个 Bash 快捷键或命令可跳到命令行中前一个单词的开头？

- a. 按 **Ctrl**+向左箭头
- b. 按 **Ctrl**+K
- c. 按 **Ctrl**+A
- d. **!string**
- e. **!number**

► 2. 哪一个 Bash 快捷键或命令可在同一行上分隔命令？

- a. 按 **Tab**
- b. **history**
- c. ;
- d. **!string**
- e. 按 **Esc**+.

► 3. 哪一个 Bash 快捷键或命令用于清除光标到命令行末尾的字符？

- a. 按 **Ctrl**+向左箭头
- b. 按 **Ctrl**+K
- c. 按 **Ctrl**+A
- d. ;
- e. 按 **Esc**+.

► 4. 哪一个 Bash 快捷键或命令用于通过匹配命令名称重新执行最近用过的命令？

- a. 按 **Tab**
- b. **!number**
- c. **!string**
- d. **history**
- e. 按 **Esc**+.

► 5. 哪一个 Bash 快捷键或命令用于补全命令、文件名和选项？

- a. ;
- b. **!number**
- c. **history**
- d. 按 **Tab**
- e. 按 **Esc**+.

► 6. 哪一个 Bash 快捷键或命令可重新执行历史记录列表中的特定命令?

- a. 按 Tab
- b. !number
- c. !string
- d. history
- e. 按 Esc+.

► 7. 哪一个 Bash 快捷键或命令可跳到命令行的开头?

- a. !number
- b. !string
- c. 按 Ctrl+ 向左箭头
- d. 按 Ctrl+K
- e. 按 Ctrl+A

► 8. 哪一个 Bash 快捷键或命令可以显示以前命令的列表?

- a. 按 Tab
- b. !string
- c. !number
- d. history
- e. 按 Esc+.

► 9. 哪一个 Bash 快捷键或命令可以复制以前命令的最后一个参数?

- a. 按 Ctrl+K
- b. 按 Ctrl+A
- c. !number
- d. 按 Esc+.

► 解决方案

使用 BASH SHELL 执行命令

选择以下问题的正确答案：

► 1. 哪一个 Bash 快捷键或命令可跳到命令行中前一个单词的开头？

- a. 按 **Ctrl+←** 向左箭头
- b. 按 **Ctrl+K**
- c. 按 **Ctrl+A**
- d. **!string**
- e. **!number**

► 2. 哪一个 Bash 快捷键或命令可在同一行上分隔命令？

- a. 按 **Tab**
- b. **history**
- c. ;
- d. **!string**
- e. 按 **Esc+.**

► 3. 哪一个 Bash 快捷键或命令用于清除光标到命令行末尾的字符？

- a. 按 **Ctrl+←** 向左箭头
- b. 按 **Ctrl+K**
- c. 按 **Ctrl+A**
- d. ;
- e. 按 **Esc+.**

► 4. 哪一个 Bash 快捷键或命令用于通过匹配命令名称重新执行最近用过的命令？

- a. 按 **Tab**
- b. **!number**
- c. **!string**
- d. **history**
- e. 按 **Esc+.**

► 5. 哪一个 Bash 快捷键或命令用于补全命令、文件名和选项？

- a. ;
- b. **!number**
- c. **history**
- d. 按 **Tab**
- e. 按 **Esc+.**

► 6. 哪一个 Bash 快捷键或命令可重新执行历史记录列表中的特定命令?

- a. 按 Tab
- b. !number
- c. !string
- d. history
- e. 按 Esc+.

► 7. 哪一个 Bash 快捷键或命令可跳到命令行的开头?

- a. !number
- b. !string
- c. 按 Ctrl+ 向左箭头
- d. 按 Ctrl+K
- e. 按 Ctrl+A

► 8. 哪一个 Bash 快捷键或命令可以显示以前命令的列表?

- a. 按 Tab
- b. !string
- c. !number
- d. history
- e. 按 Esc+.

► 9. 哪一个 Bash 快捷键或命令可以复制以前命令的最后一个参数?

- a. 按 Ctrl+K
- b. 按 Ctrl+A
- c. !number
- d. 按 Esc+.

► 开放研究实验

访问命令行

任务执行清单

在本实验中，您将使用 Bash shell 来执行命令。

成果

- 使用 Bash shell 命令行成功运行简单的程序。
- 执行用于识别文件类型并显示文本文件部分内容的命令。
- 练习使用一些 Bash 命令历史记录“快捷键”来更高效地重复命令或部分命令。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab cli-review start** 脚本来设置干净的实验环境。该脚本还会将 **zcat** 文件复制到 student 的主目录。

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

1. 使用 **date** 命令来显示当前的日期和时间。
2. 以 12 小时制显示当前时间（例如，11:42:11 AM）。提示：显示该输出的格式字符串为 **%r**。
3. **/home/student/zcat** 的文件类型是什么？可否被人读取？
4. 使用 **wc** 命令和 Bash 快捷键显示 **zcat** 的大小。
5. 显示 **zcat** 的前 10 行。
6. 显示 **zcat** 文件的后 10 行。
7. 利用三次或更少的击键来完全重复上一个命令。
8. 重复上一命令，但使用 **-n 20** 选项来显示文件中的最后 20 行。使用命令行编辑功能，以通过最少的击键次数来完成此操作。
9. 使用 shell 历史记录来再次运行 **date +%r** 命令。

评估

在 workstation 上，运行 **lab cli-review grade** 脚本来确认是否成功完成本练习。

```
[student@workstation ~]$ lab cli-review grade
```

完成

在 workstation 上，运行 **lab cli-review finish** 脚本来完成本实验。

```
[student@workstation ~]$ lab cli-review finish
```

本实验到此结束。

► 解决方案

访问命令行

任务执行清单

在本实验中，您将使用 Bash shell 来执行命令。

成果

- 使用 Bash shell 命令行成功运行简单的程序。
- 执行用于识别文件类型并显示文本文件部分内容的命令。
- 练习使用一些 Bash 命令历史记录“快捷键”来更高效地重复命令或部分命令。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab cli-review start** 脚本来设置干净的实验环境。该脚本还会将 **zcat** 文件复制到 student 的主目录。

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

1. 使用 **date** 命令来显示当前的日期和时间。

```
[student@workstation ~]$ date  
Thu Jan 22 10:13:04 PDT 2019
```

2. 以 12 小时制显示当前时间（例如，11:42:11 AM）。提示：显示该输出的格式字符串为 **%r**。
使用 **+%r** 参数运行 **date** 命令，以 12 小时制显示当前的时间。

```
[student@workstation ~]$ date +%r  
10:14:07 AM
```

3. **/home/student/zcat** 的文件类型是什么？可否被人读取？
使用 **file** 命令确定其文件类型。

```
[student@workstation ~]$ file zcat  
zcat: POSIX shell script, ASCII text executable
```

4. 使用 **wc** 命令和 Bash 快捷键显示 **zcat** 的大小。
wc 命令可用于显示 **zcat** 脚本中的行数、字数和字节数。使用 Bash 历史记录快捷键 **Esc+. (同时按 Esc 和 . 键)** 来重用上一命令中的参数，而不是重新键入文件名。

```
[student@workstation ~]$ wc Esc+.  
[student@workstation ~]$ wc zcat  
51 299 1983 zcat
```

5. 显示 **zcat** 的前 10 行。

head 命令显示文件的开头。再次尝试使用 **Esc+.** 快捷键。

```
[student@workstation ~]$ head Esc+.  
[student@workstation ~]$ head zcat  
#!/bin/sh  
# Uncompress files to standard output.  
  
# Copyright (C) 2007, 2010-2018 Free Software Foundation, Inc.  
  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU General Public License as published by  
# the Free Software Foundation; either version 3 of the License, or  
# (at your option) any later version.
```

6. 显示 **zcat** 文件的后 10 行。

使用 **tail** 命令显示 **zcat** 文件的后 10 行。

```
[student@workstation ~]$ tail Esc+.  
[student@workstation ~]$ tail zcat  
With no FILE, or when FILE is -, read standard input.  
  
Report bugs to <bug-gzip@gnu.org>."  
  
case $1 in  
--help) printf '%s\n' "$usage" || exit 1;;  
--version) printf '%s\n' "$version" || exit 1;;  
esac  
  
exec gzip -cd "$@"
```

7. 利用三次或更少的击键来完全重复上一个命令。

完全重复上一个命令。按 **向上箭头** 键一次从命令历史记录中往前回滚一个命令，然后按 **Enter** 键（使用两次击键），或者输入快捷键命令 **!!**，然后按 **Enter** 键（使用三次击键）以运行命令历史记录中的最近一个命令。（两种方法都试一次。）

```
[student@workstation]$ !!  
tail zcat  
With no FILE, or when FILE is -, read standard input.  
  
Report bugs to <bug-gzip@gnu.org>."  
  
case $1 in  
--help) printf '%s\n' "$usage" || exit 1;;
```

```
--version) printf '%s\n' "$version" || exit 1;;
esac

exec gzip -cd "$@"
```

8. 重复上一命令，但使用 **-n 20** 选项来显示文件中的最后 20 行。使用命令行编辑功能，以通过最少的击键次数来完成此操作。

向上箭头 可显示上一个命令。**Ctrl+A** 可使光标跳到行首。**Ctrl+向右箭头** 可跳到下一个字，然后添加 **-n 20** 选项并按 **Enter** 以执行该命令。

```
[student@workstation ~]$ tail -n 20 zcat
-l, --list      list compressed file contents
-q, --quiet     suppress all warnings
-r, --recursive operate recursively on directories
-S, --suffix=SUF use suffix SUF on compressed files
--synchronous   synchronous output (safer if system crashes, but slower)
-t, --test       test compressed file integrity
-v, --verbose    verbose mode
--help          display this help and exit
--version       display version information and exit
```

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>."

```
case $1 in
--help)   printf '%s\n' "$usage"  || exit 1; exit;;
--version) printf '%s\n' "$version" || exit 1; exit;;
esac

exec gzip -cd "$@"
```

9. 使用 shell 历史记录来再次运行 **date +%r** 命令。

使用 **history** 命令显示之前命令的列表，以确定要执行的具体 **date** 命令。使用 **!number** 运行命令，其中 number 自 **history** 命令输出中取用的命令编号。

请注意，您的 shell 历史记录可能与以下示例不同。根据自己的 **history** 命令的输出，确定要使用的命令编号。

```
[student@workstation ~]$ history
1 date
2 date +%r
3 file zcat
4 wc zcat
5 head zcat
6 tail zcat
7 tail -n 20 zcat
8 history
[student@workstation ~]$ !2
date +%r
10:49:56 AM
```

评估

在 **workstation** 上，运行 lab cli-review grade 脚本来确认是否成功完成本练习。

```
[student@workstation ~]$ lab cli-review grade
```

完成

在 workstation 上，运行 **lab cli-review finish** 脚本来完成本实验。

```
[student@workstation ~]$ lab cli-review finish
```

本实验到此结束。

总结

在本章中，您学到了：

- Bash shell 是一个命令解释器，它提示交互用户指定 Linux 命令。
- 很多命令都有一个 **--help** 选项，可显示用法消息或屏幕。
- 使用工作区可以更轻松地组织多个应用窗口。
- 位于顶栏左上角的 **Activities** 按钮可提供一个概览模式，帮护用户组织窗口并启动应用。
- **file** 命令可以扫描文件内容的开头，显示该文件的类型。
- **head** 和 **tail** 命令分别显示文件的开头和结尾部分。
- 您可以使用 **Tab** 补全在键入文件名作为命令的参数时将它们补全。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

章 3

从命令行管理文件

目标

在从 Bash shell 操作时，复制、移动、创建、删除和组织文件。

培训目标

- 描述 Linux 如何组织文件，以及文件系统层次结构中各种目录的用途。
- 指定文件相对于当前工作目录的位置和绝对位置，确定并更改工作目录，以及列出目录的内容。
- 创建、复制、移动和删除文件及目录。
- 使用硬链接和符号（或“软”）链接，使多个文件名引用同一文件。
- 通过使用 Bash shell 的模式匹配功能，高效地运行影响很多文件的命令。

章节

- 描述 Linux 文件系统层次结构概念（及测验）
- 通过名称指定文件（及测验）
- 使用命令行工具管理文件（及引导式练习）
- 制作文件间的链接（及引导式练习）
- 使用 Shell 扩展匹配文件名（及测验）

实验

从命令行管理文件

描述 LINUX 文件系统层次结构概念

培训目标

学完本节后，您应能够描述 Linux 如何组织文件，以及文件系统层次结构中各种目录的用途。

文件系统层次结构

Linux 系统中的所有文件存储在文件系统中，它们被组织到一个颠倒的目录树中，称为文件系统层次结构。这棵树是颠倒的，因为树根在该层次结构的顶部，树根的下方延伸出目录和子目录的分支。

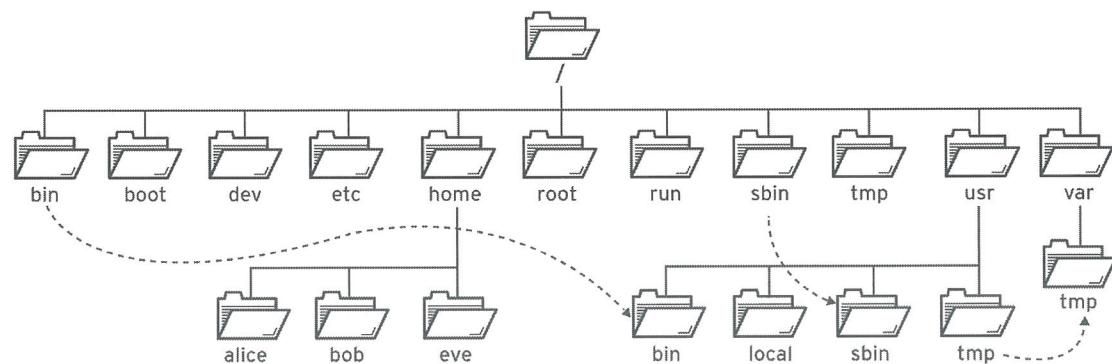


图 3.1: 红帽企业 Linux 8 中的重要文件系统目录

/ 目录是根目录，位于文件系统层次结构的顶部。/ 字符还用作文件名中的目录分隔符。例如，如果 **etc** 是 / 目录的子目录，您可以将该目录指代为 **/etc**。类似地，如果 **/etc** 目录包含一个名为 **issue** 的文件，您可以将该文件指代为 **/etc/issue**。

/ 的子目录用于标准化的用途，以便根据文件和用途组织文件。这可以方便查找文件。例如，在根目录中，子目录 **/boot** 用于存储启动系统所需的文件。



注意

下列术语有助于描述文件系统目录内容：

- 静态内容将保持不变，除非以显式方式编辑或重新配置。
- 动态或可变内容可由活动进程进行修改或附加。
- 永久内容会在重启后依然保留，例如配置设置。
- 运行时内容是在重启时会被删除的流程或系统特定内容。

下表根据名称和用途列出了系统中一些最重要的目录。

红帽企业 Linux 重要目录

位置	用途
/usr	安装的软件、共享的库，包括文件和只读程序数据。重要的子目录有： · /usr/bin : 用户命令。 · /usr/sbin : 系统管理命令。 · /usr/local : 本地自定义软件。
/etc	特定于此系统的配置文件。
/var	特定于此系统的可变数据，在系统启动之间保持永久性。动态变化的文件（如数据库、缓存目录、日志文件、打印机后台处理文档和网站内容）可以在 /var 下找到。
/run	自上一次系统启动以来启动的进程的运行时数据。这包括进程 ID 文件和锁定文件，等等。此目录中的内容在重启时重新创建。此目录合并了早期版本的红帽企业 Linux 中的 /var/run 和 /var/lock 。
/home	主目录是普通用户存储其个人数据和配置文件的位置。
/root	管理超级用户 root 的主目录。
/tmp	供临时文件使用的全局可写空间。10 天内未访问、未更改或未修改的文件将自动从该目录中删除。还有一个临时目录 /var/tmp ，该目录中的文件如果在 30 天内未曾访问、更改或修改过，将被自动删除。
/boot	开始启动过程所需的文件。
/dev	包含特殊的设备文件，供系统用于访问硬件。



重要

在红帽企业 Linux 7 及更高版本中，**/** 中的四个较旧的目录现在与它们在 **/usr** 中对应的目录拥有完全相同的内容：

- **/bin** 和 **/usr/bin**
- **/sbin** 和 **/usr/sbin**
- **/lib** 和 **/usr/lib**
- **/lib64** 和 **/usr/lib64**

在红帽企业 Linux 的较早版本中，这些是不同的目录，包含几组不同的文件。

在红帽企业 Linux 7 及更高版本中，**/** 中的目录是 **/usr** 中对应目录的符号链接。



参考文献

hier(7) 手册

Fedora 17 的 UsrMove 功能页

<https://fedoraproject.org/wiki/Features/UsrMove>

► 小测验

描述 LINUX 文件系统层次结构概念

选择以下问题的正确答案：

► 1. 哪一个目录包含永久的系统特定配置数据？

- a. /etc
- b. /root
- c. /run
- d. /usr

► 2. 哪一个目录位于文件系统层次结构的顶部？

- a. /etc
- b. /
- c. /home/root
- d. /root

► 3. 哪一个目录包含用户主目录？

- a. /
- b. /home
- c. /root
- d. /user

► 4. 哪一个目录包含临时文件？

- a. /tmp
- b. /trash
- c. /run
- d. /var

► 5. 哪一个目录包含数据库和网站等的动态数据？

- a. /etc
- b. /run
- c. /usr
- d. /var

► 6. 哪一个目录是管理超级用户的主目录？

- a. /etc
- b. /
- c. /home/root
- d. /root

► 7. 哪一个目录包含普通命令和实用程序?

- a. /commands
- b. /run
- c. /usr/bin
- d. /usr/sbin

► 8. 哪一个目录包含非永久的进程运行时数据?

- a. /tmp
- b. /etc
- c. /run
- d. /var

► 9. 哪一个目录包含已安装的软件程序和库?

- a. /etc
- b. /lib
- c. /usr
- d. /var

► 解决方案

描述 LINUX 文件系统层次结构概念

选择以下问题的正确答案：

► 1. 哪一个目录包含永久的系统特定配置数据？

- a. /etc
- b. /root
- c. /run
- d. /usr

► 2. 哪一个目录位于文件系统层次结构的顶部？

- a. /etc
- b. /
- c. /home/root
- d. /root

► 3. 哪一个目录包含用户主目录？

- a. /
- b. /home
- c. /root
- d. /user

► 4. 哪一个目录包含临时文件？

- a. /tmp
- b. /trash
- c. /run
- d. /var

► 5. 哪一个目录包含数据库和网站等的动态数据？

- a. /etc
- b. /run
- c. /usr
- d. /var

► 6. 哪一个目录是管理超级用户的主目录？

- a. /etc
- b. /
- c. /home/root
- d. /root

► 7. 哪一个目录包含普通命令和实用程序?

- a. /commands
- b. /run
- c. /usr/bin
- d. /usr/sbin

► 8. 哪一个目录包含非永久的进程运行时数据?

- a. /tmp
- b. /etc
- c. /run
- d. /var

► 9. 哪一个目录包含已安装的软件程序和库?

- a. /etc
- b. /lib
- c. /usr
- d. /var

通过名称指定文件

培训目标

学完本节后，您应能够指定文件相对于当前工作目录的位置和绝对位置，确定并更改工作目录，以及列出目录的内容。

绝对路径和相对路径

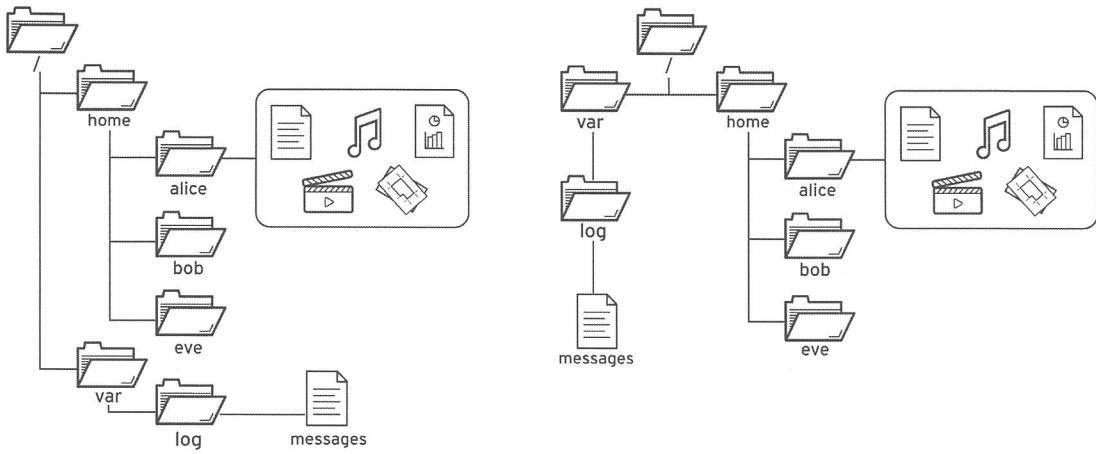


图 3.2: 常用文件浏览器视图（左）与自上而下视图（右）相当。

文件或目录的路径指定其唯一的文件系统位置。跟随文件路径会遍历一个或多个指定的子目录，用正斜杠 (/) 分隔，直到到达目标位置。目录也称为文件夹，包含其他文件和其他子目录。它们的引用方式可以与文件相同。



重要

可以接受空格字符作为 Linux 文件名的一部分。但是，shell 也使用空格来分隔命令行上的选项和参数。如果您输入的命令中包含了名称带有空格的文件，则 shell 可能会误解该命令，并认为您要在空格处开始一个新文件名或其他参数。

通过将文件名放在引号中，可以避免这种情况。但如果并不需要在文件名中使用空格，直接避免使用空格会更简单。

绝对路径

绝对路径是一个完全限定的名称，用于指定文件在文件系统层次结构中的确切位置。它从根 (/) 目录开始，并指定为到达特定文件所必须遍历的每个子目录。文件系统中的每个文件都有一个唯一绝对路径名，可通过一个简单的规则识别：第一个字符是正斜杠 (/) 的路径名是绝对路径名。例如，系统消息日志文件的绝对路径名是 **/var/log/messages**。绝对路径名键入起来可能会太长，所以也可通过相对于 shell 提示符当前工作目录的位置来查找文件。

当前工作目录和相对路径

当用户登录并打开命令窗口时，初始位置通常是该用户的主目录。系统进程也有初始目录。用户和进程根据需要导航至其他目录；术语工作目录或当前工作目录指它们的当前位置。

与绝对路径一样，相对路径也标识唯一文件，仅指定从工作目录到达该文件所需的路径。识别相对路径名遵循一个简单规则：第一个字符是正斜杠之外的其他字符的路径名是相对路径名。位于 **/var** 目录的用户可以将消息日志文件相对指代为 **log/messages**。

Linux 文件系统，包含但不限于 ext4、XFS、GFS2 和 GlusterFS，都是区分大小写的。在同一目录中创建 **FileCase.txt** 和 **filecase.txt** 将生成两个不同的文件。

非 Linux 文件系统的工作方式可能会有所不同。例如，VFAT、Microsoft 的 NTFS 和 Apple 的 HFS+ 具有大小写保留行为。虽然这些文件系统不区分大小写，它们的确使用文件创建时的原始大小写形式显示文件名。因此，如果您试图在 VFAT 文件系统上创建上述示例中的文件，那么这两个名称将被视为指向同一文件，而不是指向两个不同的文件。

导航路径

pwd 命令显示该 shell 的当前工作目录的完整路径名。这可以帮助您确定使用相对路径名来访问文件的语法。**ls** 命令列出指定目录的目录内容；如果未指定目录，则列出当前工作目录的内容。

```
[user@host ~]$ pwd
/home/user
[user@host ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[user@host ~]$
```

使用 **cd** 命令可更改 shell 的当前工作目录。如果没有为该命令指定任何参数，它将切换到您的主目录。

在下面的示例中，**cd** 命令混用了绝对路径和相对路径，以更改 shell 的当前工作目录。

```
[user@host ~]$ pwd
/home/user
[user@host ~]$ cd Videos
[user@host Videos]$ pwd
/home/user/Videos
[user@host Videos]$ cd /home/user/Documents
[user@host Documents]$ pwd
/home/user/Documents
[user@host Documents]$ cd
[user@host ~]$ pwd
/home/user
[user@host ~]$
```

正如您在上述示例中所看到的，默认 shell 提示符还显示当前工作目录绝对路径的最后一个组件。例如，对于 **/home/user/Videos**，仅显示 **Videos**。当您的当前工作目录是主目录时，提示符显示波形符字符 (**~**)。

touch 该命令通常将文件的时间戳更新为当前日期和时间，而不进行其他修改。这通常可用于创建空文件（可供练习使用），因为“触碰”不存在的文件名会导致创建该文件。在以下示例中，**touch** 命令在 **Documents** 和 **Videos** 子目录中创建练习文件。

```
[user@host ~]$ touch Videos/blockbuster1.ogg
[user@host ~]$ touch Videos/blockbuster2.ogg
[user@host ~]$ touch Documents/thesis_chapter1.odf
[user@host ~]$ touch Documents/thesis_chapter2.odf
[user@host ~]$
```

ls 命令具有多个选项，用于显示文件的属性。最常见且最有用的选项是 **-l**（长列表格式）、**-a**（包含隐藏文件在内的所有文件），以及 **-R**（递归方式，包含所有子目录的内容）。

```
[user@host ~]$ ls -l
total 15
drwxr-xr-x. 2 user user 4096 Feb 7 14:02 Desktop
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Documents
drwxr-xr-x. 3 user user 4096 Jan 9 15:00 Downloads
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Music
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Pictures
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Public
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Templates
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Videos
[user@host ~]$ ls -la
total 15
drwx----- 16 user user 4096 Feb 8 16:15 .
drwxr-xr-x. 6 root root 4096 Feb 8 16:13 ..
-rw----- 1 user user 22664 Feb 8 00:37 .bash_history
-rw-r--r--. 1 user user 18 Jul 9 2013 .bash_logout
-rw-r--r--. 1 user user 176 Jul 9 2013 .bash_profile
-rw-r--r--. 1 user user 124 Jul 9 2013 .bashrc
drwxr-xr-x. 4 user user 4096 Jan 20 14:02 .cache
drwxr-xr-x. 8 user user 4096 Feb 5 11:45 .config
drwxr-xr-x. 2 user user 4096 Feb 7 14:02 Desktop
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Documents
drwxr-xr-x. 3 user user 4096 Jan 25 20:48 Downloads
drwxr-xr-x. 11 user user 4096 Feb 6 13:07 .gnome2
drwx----- 2 user user 4096 Jan 20 14:02 .gnome2_private
-rw----- 1 user user 15190 Feb 8 09:49 .ICEauthority
drwxr-xr-x. 3 user user 4096 Jan 9 15:00 .local
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Music
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Pictures
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Public
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Templates
drwxr-xr-x. 2 user user 4096 Jan 9 15:00 Videos
[user@host ~]$
```

列表顶部的两个特殊目录是当前目录(.)和父目录(..)。这些特殊目录存在于系统中的每一目录。开始使用文件管理命令时，您将发现它们的用处。