

```
[user@host glob]$ echo file{a,b}{1,2}.txt  
filea1.txt filea2.txt fileb1.txt fileb2.txt  
[user@host glob]$ echo file{a{1,2},b,c}.txt  
filea1.txt filea2.txt fileb.txt filec.txt  
[user@host glob]$
```

大括号扩展的实际用途是快速创建多个文件或目录。

```
[user@host glob]$ mkdir .../RHEL{6,7,8}  
[user@host glob]$ ls .../RHEL*  
RHEL6 RHEL7 RHEL8  
[user@host glob]$
```

变量扩展

变量的作用类似于可以在内存中存储值的命名容器。通过变量，可以从命令行或在 shell 脚本内轻松访问和修改存储的数据。

您可以通过以下语法将数据作为值分配给变量：

```
[user@host ~]$ VARIABLENAME=value
```

可以使用变量扩展将变量名称转换为命令行上的值。如果字符串以美元符号 (\$) 开头，那么 shell 就会尝试将该字符串的其余部分用作变量名称，并将它替换为变量中包含的任何值。

```
[user@host ~]$ USERNAME=operator  
[user@host ~]$ echo $USERNAME  
operator
```

为了避免因其他 shell 扩展而引起的错误，您可以将变量的名称放在大括号中，如 \${VARIABLENAME}。

```
[user@host ~]$ USERNAME=operator  
[user@host ~]$ echo ${USERNAME}  
operator
```

Shell 变量及其使用方式将在本课程的后续部分进行更深入的介绍。

命令替换

命令替换允许命令的输出替换命令行上的命令本身。当命令括在括号中并且前面有美元符号 (\$) 时，会发生命令替换。\$(命令) 形式可以互相嵌套多个命令扩展。

```
[user@host glob]$ echo Today is $(date +%A).  
Today is Wednesday.  
[user@host glob]$ echo The time is $(date +%M) minutes past $(date +%l%p).  
The time is 26 minutes past 11AM.  
[user@host glob]$
```

**注意**

较旧形式的命令替换使用反引号：`命令`。反引号形式的缺点包括：1) 反引号在视觉上很容易与单引号混淆；2) 反引号无法嵌套。

防止参数被扩展

在 Bash shell 中，许多字符有特殊含义。为了防止 shell 在命令行的某些部分上执行 shell 扩展，您可以为字符和字符串加引号或执行转义。

反斜杠 (\) 是 Bash shell 中的转义字符。它可以防止紧随其后的字符被扩展。

```
[user@host glob]$ echo The value of $HOME is your home directory.
The value of /home/user is your home directory.
[user@host glob]$ echo The value of \$HOME is your home directory.
The value of $HOME is your home directory.
[user@host glob]$
```

在上面的示例中，保护美元符号免于扩展导致 Bash 将其视为常规字符，因此也就未在 **\$HOME** 上执行变量扩展。

如果要保护较长的字符串，则使用单引号 ('') 或双引号 ("") 来括起字符串。它们的作用略有不同。单引号将阻止所有 shell 扩展。双引号则阻止大部分 shell 扩展。

使用双引号可以阻止通配和 shell 扩展，但依然允许命令和变量替换。

```
[user@host glob]$ myhost=$(hostname -s); echo $myhost
host
[user@host glob]$ echo "***** hostname is ${myhost} *****"
***** hostname is host *****
[user@host glob]$
```

使用单引号则可以按字面解译所有的文本。

```
[user@host glob]$ echo "Will variable $myhost evaluate to $(hostname -s)?"
Will variable myhost evaluate to host?
[user@host glob]$ echo 'Will variable $myhost evaluate to $(hostname -s)?'
Will variable $myhost evaluate to $(hostname -s)?
[user@host glob]$
```

**重要**

无论是在屏幕还是在键盘上，单引号 ('') 和命令替换反引号 (`) 都很容易混淆。不恰当的意图的使用可导致意外的 shell 行为。

**参考文献**

bash(1)、**cd(1)**、**glob(7)**、**isalpha(3)**、**ls(1)**、**path_resolution(7)** 和 **pwd(1)**
man page

► 小测验

使用 SHELL 扩展匹配文件名

选择以下问题的正确答案：

► 1. 哪一种模式仅匹配以“b”结尾的文件名？

- a. **b***
- b. ***b**
- c. ***b***
- d. **[!b]***

► 2. 哪一种模式仅匹配以“b”开头的文件名？

- a. **b***
- b. ***b**
- c. ***b***
- d. **[!b]***

► 3. 哪一种模式仅匹配首字符不是“b”的文件名？

- a. **b***
- b. ***b**
- c. ***b***
- d. **[!b]***

► 4. 哪一种模式匹配包含“b”的所有文件名？

- a. **b***
- b. ***b**
- c. ***b***
- d. **[!b]***

► 5. 哪一种模式仅匹配包含数字的文件名？

- a. ***#***
- b. ***[[digit:]]***
- c. ***[digit]***
- d. **[0-9]**

► 6. 哪一种模式仅匹配以大写字母开头的文件名?

- a. ^?*
- b. ^*
- c. [upper]*
- d. [[:upper:]]*
- e. [[CAP]]*

► 7. 哪一种模式仅匹配长度至少为三个字符的文件名?

- a. ???*
- b. ???
- c. \3*
- d. +++*
- e.*

► 解决方案

使用 SHELL 扩展匹配文件名

选择以下问题的正确答案：

► 1. 哪一种模式仅匹配以“b”结尾的文件名？

- a. b*
- b. *b
- c. *b*
- d. [!b]*

► 2. 哪一种模式仅匹配以“b”开头的文件名？

- a. b*
- b. *b
- c. *b*
- d. [!b]*

► 3. 哪一种模式仅匹配首字符不是“b”的文件名？

- a. b*
- b. *b
- c. *b*
- d. [!b]*

► 4. 哪一种模式匹配包含“b”的所有文件名？

- a. b*
- b. *b
- c. *b*
- d. [!b]*

► 5. 哪一种模式仅匹配包含数字的文件名？

- a. *#*
- b. *[:digit:]*
- c. *[digit]*
- d. [0-9]

► 6. 哪一种模式仅匹配以大写字母开头的文件名?

- a. ^?*
- b. ^*
- c. [upper]*
- d. [[:upper:]]*
- e. [[CAP]]*

► 7. 哪一种模式仅匹配长度至少为三个字符的文件名?

- a. ???*
- b. ???
- c. \3*
- d. +++*
- e. *** *

► 开放研究实验

从命令行管理文件

任务执行清单

在本实验中，您将通过利用 shell 和各种文件名匹配技巧来高效地创建、移动和删除文件与目录。

成果

您应能够：

- 使用通配符来定位和操作文件。

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 files-review start** 命令。此命令将运行一个起始脚本，它将确定 serverb 计算机是否可从网络访问。

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

1. 使用 **ssh** 命令，以 student 用户身份登录 serverb。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. 在创建项目文件之前，请使用 **mkdir** 命令及大括号扩展在 **/home/student/Documents/project_plans** 目录中创建空的项目计划文档。（提示：如果 **~/Documents** 不存在，**mkdir** 命令的 **-p** 选项会创建它）
在 **~/Documents/project_plans** 目录中创建以下两个空文件：**season1_project_plan.odf** 和 **season2_project_plan.odf**。
3. 为本实验创建一套空白的练习文件。如果无法立即识别所需的 shell 扩展快捷键，请使用答案来学习和练习。使用 shell tab 补全功能可以轻松查找文件路径名。
创建名为 **tv_seasonX_episodeY.ogg** 的文件，一共 12 个。将 X 替换为季数，Y 替换为该季中的集数，共两季，每季六集。
4. 作为一名成功的连载悬疑小说作者，您正在为出版下一部畅销书进行编辑。创建名为 **mystery_chapterX.odf** 的文件，一共 8 个。将 X 替换为数字 1 到 8。
5. 使用一个命令在 **season1** 目录下创建两个子目录，名称为 **season2** 和 **Videos**，以整理电视剧集。
6. 将适当的电视剧集移到电视季子目录中。仅可用两个命令，使用相对语法指定目的地。

7. 利用一个命令创建两级目录层次结构，以整理悬疑小说章节。在 **Documents** 目录下创建 **my_bestseller**，并在新的 **my_bestseller** 目录下创建 **chapters**。
8. 利用一个命令，直接在 **my_bestseller** 目录下再创建 3 个子目录。将这些子目录命名为 **editor**、**changes** 和 **vacation**。不需要 **-p** 选项（创建父级），因为 **my_bestseller** 父目录已经存在。
9. 更改为 **chapters** 目录。使用波形符 (~) 主目录快捷键指定源文件，将所有图书章节移到 **chapters** 目录中，即您现在的当前目录。指定目标目录的最简单语法是什么？
10. 您已将前两个章节将发送给编辑进行审阅。仅将这两个章节移到 **editor** 目录，以避免在审阅期间修改它们。从章节子目录开始，使用带有范围的大括号扩展来指定要移动的章节文件名和目标目录的相对路径。
11. 在度假期间，您打算编写第 7 和第 8 章。使用一个命令将文件从 **chapters** 目录移到 **vacation** 目录。使用带有字符串列表的大括号扩展指定章节文件名，而不使用通配符字符。
12. 将您的工作目录更改为 **~/Videos/season2**，然后将本季的第一集复制到 **vacation** 目录。
13. 使用一个 **cd** 命令，从工作目录更改为 **~/Documents/my_bestseller/vacation** 目录。列出其文件。使用上一工作目录参数，返回到 **season2** 目录。（如果最近一次用 **cd** 命令更改目录是通过一个命令而非多个 **cd** 命令来完成的，这将会成功。）从 **season2** 目录，将第 2 集文件复制到 **vacation** 目录。再次使用快捷键返回到 **vacation** 目录。
14. 第 5 章和第 6 章的作者想要试验可能的修订。将这两个文件从 **~/Documents/my_bestseller/chapters** 目录复制到 **~/Documents/my_bestseller/changes** 目录，以防止这些修订会修改原始文件。前往 **~/Documents/my_bestseller** 目录。使用方括号模式匹配来指定要在 **cp** 命令的文件名参数中匹配的章节号。
15. 将当前工作目录更改到 **changes** 目录。
通过命令替换使用 **date +%F** 命令将 **mystery_chapter5.odf** 复制到包含完整日期的新文件。名称应该采用 **mystery_chapter5_YYYY-MM-DD.odf** 形式。
制作 **mystery_chapter5.odf** 的另一个副本，并且附上当前的时间戳（即自该时期 1970-01-01 00:00 UTC 起的秒数），确保文件名独一无二。使用命令替换及 **date +%s** 命令来实现这一目标。
16. 进一步审阅后，您决定不需要修订情节。删除 **changes** 目录。
如有必要，前往 **changes** 目录并删除该目录中的所有文件。如果是当前工作目录，您无法删除该目录。更改到 **changes** 目录的父目录。尝试使用 **rm** 命令（不带 **-r** 递归选项）删除空目录。此尝试应该会失败。最后，使用 **rmdir** 命令删除空目录，这将成功。
17. 度假结束后，不再需要 **vacation** 目录。使用 **rm** 命令（带递归选项）删除该目录。
完成时，返回到 **student** 用户的主目录。
18. 创建 **~/Documents/project_plans/season2_project_plan.odf** 文件的硬链接，取名为 **~/Documents/backups/season2_project_plan.odf.back**。硬链接将防止意外删除原始文件，并在更改原始文件时使备份文件保持更新。
19. 从 **serverb** 退出。

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

评估

在 workstation 上，运行 **lab files-review grade** 脚本来确认是否成功完成本实验。

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

完成

在 workstation 上，运行 **lab files-review finish** 脚本来结束本实验。此脚本将删除实验练习过程中在 serverb 上创建的所有文件和目录。

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

本实验到此结束。

▶ 解决方案

从命令行管理文件

任务执行清单

在本实验中，您将通过利用 shell 和各种文件名匹配技巧来高效地创建、移动和删除文件与目录。

成果

您应能够：

- 使用通配符来定位和操作文件。

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 files-review start** 命令。此命令将运行一个起始脚本，它将确定 serverb 计算机是否可从网络访问。

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

1. 使用 **ssh** 命令，以 student 用户身份登录 serverb。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. 在创建项目文件之前，请使用 **mkdir** 命令及大括号扩展在 **/home/student/Documents/project_plans** 目录中创建空的项目计划文档。（提示：如果 **~/Documents** 不存在，**mkdir** 命令的 **-p** 选项会创建它。）

在 **~/Documents/project_plans** 目录中创建以下两个空文件：**season1_project_plan.odf** 和 **season2_project_plan.odf**。

```
[student@serverb ~]$ mkdir -p ~/Documents/project_plans  
[student@serverb ~]$ touch \  
~/Documents/project_plans/{season1,season2}_project_plan.odf  
[student@serverb ~]$ ls -lR Documents/  
Documents/:  
total 0  
drwxrwxr-x. 2 student student 70 Jan 31 18:20 project_plans  
  
Documents/project_plans:  
total 0  
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1_project_plan.odf  
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season2_project_plan.odf
```

3. 为本实验创建一套空白的练习文件。如果无法立即识别所需的 shell 扩展快捷键，请使用答案来学习和练习。使用 shell tab 补全功能可以轻松查找文件路径名。

创建名为 **tv_seasonX_episodeY.ogg** 的文件，一共 12 个。将 X 替换为季数，Y 替换为该季中的集数，共两季，每季六集。

```
[student@serverb ~]$ touch tv_season{1..2}_episode{1..6}.ogg  
[student@serverb ~]$ ls tv*  
tv_season1_episode1.ogg  tv_season1_episode5.ogg  tv_season2_episode3.ogg  
tv_season1_episode2.ogg  tv_season1_episode6.ogg  tv_season2_episode4.ogg  
tv_season1_episode3.ogg  tv_season2_episode1.ogg  tv_season2_episode5.ogg  
tv_season1_episode4.ogg  tv_season2_episode2.ogg  tv_season2_episode6.ogg
```

4. 作为一名成功的连载悬疑小说作者，您正在为出版下一部畅销书进行编辑。创建名为 **mystery_chapterX.odf** 的文件，一共 8 个。将 X 替换为数字 1 到 8。

```
[student@serverb ~]$ touch mystery_chapter{1..8}.odf  
[student@serverb ~]$ ls mys*  
mystery_chapter1.odf  mystery_chapter4.odf  mystery_chapter7.odf  
mystery_chapter2.odf  mystery_chapter5.odf  mystery_chapter8.odf  
mystery_chapter3.odf  mystery_chapter6.odf
```

5. 使用一个命令在 **season1** 目录下创建两个子目录，名称为 **season2** 和 **Videos**，以整理电视剧集。

```
[student@serverb ~]$ mkdir -p Videos/season{1..2}  
[student@serverb ~]$ ls Videos  
season1  season2
```

6. 将适当的电视剧集移到电视季子目录中。仅可用两个命令，使用相对语法指定目的地。

```
[student@serverb ~]$ mv tv_season1* Videos/season1  
[student@serverb ~]$ mv tv_season2* Videos/season2  
[student@serverb ~]$ ls -R Videos  
Videos:  
season1  season2  
  
Videos/season1:  
tv_season1_episode1.ogg  tv_season1_episode3.ogg  tv_season1_episode5.ogg  
tv_season1_episode2.ogg  tv_season1_episode4.ogg  tv_season1_episode6.ogg  
  
Videos/season2:  
tv_season2_episode1.ogg  tv_season2_episode3.ogg  tv_season2_episode5.ogg  
tv_season2_episode2.ogg  tv_season2_episode4.ogg  tv_season2_episode6.ogg
```

7. 利用一个命令创建两级目录层次结构，以整理悬疑小说章节。在 **Documents** 目录下创建 **my_bestseller**，并在新的 **my_bestseller** 目录下创建 **chapters**。

```
[student@serverb ~]$ mkdir -p Documents/my_bestseller/chapters  
[student@serverb ~]$ ls -R Documents  
Documents:  
my_bestseller  project_plans
```

```
Documents/my_bestseller:  
chapters
```

```
Documents/my_bestseller/chapters:
```

```
Documents/project_plans:  
season1_project_plan.odf season2_project_plan.odf
```

8. 利用一个命令，直接在 **my_bestseller** 目录下再创建 3 个子目录。将这些子目录命名为 **editor**、**changes** 和 **vacation**。不需要 **-p** 选项（创建父级），因为 **my_bestseller** 父目录已经存在。

```
[student@serverb ~]$ mkdir Documents/my_bestseller/{editor,changes,vacation}
```

```
[student@serverb ~]$ ls -R Documents
```

```
Documents:  
my_bestseller project_plans
```

```
Documents/my_bestseller:  
changes chapters editor vacation
```

```
Documents/my_bestseller/changes:
```

```
Documents/my_bestseller/chapters:
```

```
Documents/my_bestseller/editor:
```

```
Documents/my_bestseller/vacation:
```

```
Documents/project_plans:
```

```
season1_project_plan.odf season2_project_plan.odf
```

9. 更改为 **chapters** 目录。使用波形符 (~) 主目录快捷键指定源文件，将所有图书章节移到 **chapters** 目录中，即您现在的当前目录。指定目标目录的最简单语法是什么？

```
[student@serverb ~]$ cd Documents/my_bestseller/chapters  
[student@serverb chapters]$ mv ~/mystery_chapter* .  
[student@serverb chapters]$ ls  
mystery_chapter1.odf mystery_chapter4.odf mystery_chapter7.odf  
mystery_chapter2.odf mystery_chapter5.odf mystery_chapter8.odf  
mystery_chapter3.odf mystery_chapter6.odf
```

10. 您已将前两个章节将发送给编辑进行审阅。仅将这两个章节移到 **editor** 目录，以避免在审阅期间修改它们。从章节子目录开始，使用带有范围的大括号扩展来指定要移动的章节文件名和目标目录的相对路径。

```
[student@serverb chapters]$ mv mystery_chapter{1..2}.odf ../editor  
[student@serverb chapters]$ ls  
mystery_chapter3.odf mystery_chapter5.odf mystery_chapter7.odf  
mystery_chapter4.odf mystery_chapter6.odf mystery_chapter8.odf  
[student@serverb chapters]$ ls ../editor  
mystery_chapter1.odf mystery_chapter2.odf
```

11. 在度假期间，您打算编写第 7 和第 8 章。使用一个命令将文件从 **chapters** 目录移到 **vacation** 目录。使用带有字符串列表的大括号扩展指定章节文件名，而不使用通配符字符。

```
[student@serverb chapters]$ mv mystery_chapter{7,8}.odf ..../vacation  
[student@serverb chapters]$ ls  
mystery_chapter3.odf mystery_chapter5.odf  
mystery_chapter4.odf mystery_chapter6.odf  
[student@serverb chapters]$ ls ..../vacation  
mystery_chapter7.odf mystery_chapter8.odf
```

12. 将您的工作目录更改为 **~/Videos/season2**，然后将本季的第一集复制到 **vacation** 目录。

```
[student@serverb chapters]$ cd ~/Videos/season2  
[student@serverb season2]$ cp *episode1.ogg ~/Documents/my_bestseller/vacation
```

13. 使用一个 **cd** 命令，从工作目录更改为 **~/Documents/my_bestseller/vacation** 目录。列出其文件。使用上一工作目录参数，返回到 **season2** 目录。（如果最近一次用 **cd** 命令更改目录是通过一个命令而非多个 **cd** 命令来完成的，这将会成功。）从 **season2** 目录，将第 2 集文件复制到 **vacation** 目录。再次使用快捷键返回到 **vacation** 目录。

```
[student@serverb season2]$ cd ~/Documents/my_bestseller/vacation  
[student@serverb vacation]$ ls  
mystery_chapter7.odf mystery_chapter8.odf tv_season2_episode1.ogg  
[student@serverb vacation]$ cd -  
/home/ec2-user/Videos/season2  
[student@serverb season2]$ cp *episode2.ogg ~/Documents/my_bestseller/vacation  
[student@serverb vacation]$ cd -  
/home/ec2-user/Documents/my_bestseller/vacation  
[student@serverb vacation]$ ls  
mystery_chapter7.odf tv_season2_episode1.ogg  
mystery_chapter8.odf tv_season2_episode2.ogg
```

14. 第 5 章和第 6 章的作者想要试验可能的修订。将这两个文件从 **~/Documents/my_bestseller/chapters** 目录复制到 **~/Documents/my_bestseller/changes** 目录，以防止这些修订会修改原始文件。前往 **~/Documents/my_bestseller** 目录。使用方括号模式匹配来指定要在 **cp** 命令的文件名参数中匹配的章节号。

```
[student@serverb vacation]$ cd ~/Documents/my_bestseller  
[student@serverb my_bestseller]$ cp chapters/mystery_chapter[56].odf changes  
[student@serverb my_bestseller]$ ls chapters  
mystery_chapter3.odf mystery_chapter5.odf  
mystery_chapter4.odf mystery_chapter6.odf  
[student@serverb my_bestseller]$ ls changes  
mystery_chapter5.odf mystery_chapter6.odf
```

15. 将当前工作目录更改到 **changes** 目录。

通过命令替换使用 **date +%F** 命令将 **mystery_chapter5.odf** 复制到包含完整日期的新文件。名称应该采用 **mystery_chapter5_YYYY-MM-DD.odf** 形式。

制作 **mystery_chapter5.odf** 的另一个副本，并且附上当前的时间戳（即自该时期 1970-01-01 00:00 UTC 起的秒数），确保文件名独一无二。使用命令替换及 **date +%s** 命令来实现这一目标。

```
[student@serverb my_bestseller]$ cd changes
[student@serverb changes]$ cp mystery_chapter5.odf \
mystery_chapter5_$(date +%F).odf
[student@serverb changes]$ cp mystery_chapter5.odf \
mystery_chapter5_$(date +%s).odf
[student@serverb changes]$ ls
mystery_chapter5_1492545076.odf mystery_chapter5.odf
mystery_chapter5_2017-04-18.odf mystery_chapter6.odf
```

16. 进一步审阅后，您决定不需要修订情节。删除 **changes** 目录。

如有必要，前往 **changes** 目录并删除该目录中的所有文件。如果是当前工作目录，您无法删除该目录。更改到 **changes** 目录的父目录。尝试使用 **rm** 命令（不带 **-r** 递归选项）删除空目录。此尝试应该会失败。最后，使用 **rmdir** 命令删除空目录，这将成功。

```
[student@serverb changes]$ rm mystery*
[student@serverb changes]$ cd ..
[student@serverb my_bestseller]$ rm changes
rm: cannot remove 'changes': Is a directory
[student@serverb my_bestseller]$ rmdir changes
[student@serverb my_bestseller]$ ls
chapters editor vacation
```

17. 度假结束后，不再需要 **vacation** 目录。使用 **rm** 命令（带递归选项）删除该目录。

完成时，返回到 **student** 用户的主目录。

```
[student@serverb my_bestseller]$ rm -r vacation
[student@serverb my_bestseller]$ ls
chapters editor
[student@serverb my_bestseller]$ cd
[student@serverb ~]$
```

18. 创建 **~/Documents/project_plans/season2_project_plan.odf** 文件的硬链接，取名为 **~/Documents/backups/season2_project_plan.odf.back**。硬链接将防止意外删除原始文件，并在更改原始文件时使备份文件保持更新。

注意 **season2_project_plan.odf.back** 和 **season2_project_plan.odf** 文件的链接数都是 **2**。

```
[student@serverb ~]$ mkdir ~/Documents/backups
[student@serverb ~]$ ln ~/Documents/project_plans/season2_project_plan.odf \
~/Documents/backups/season2_project_plan.odf.back
[student@serverb ~]$ ls -lR ~/Documents/
/home/student/Documents/:
total 0
drwxrwxr-x. 2 student student 43 Jan 31 18:59 backups
```

```
drwxrwxr-x. 4 student student 36 Jan 31 19:42 my_bestseller
drwxrwxr-x. 2 student student 70 Jan 31 18:20 project_plans

/home/student/Documents/backups:
total 4
-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2_project_plan.odf.back

/home/student/Documents/my_bestseller:
total 0
drwxrwxr-x. 2 student student 118 Jan 31 19:39 chapters
drwxrwxr-x. 2 student student 62 Jan 31 19:38 editor

/home/student/Documents/my_bestseller/chapters:
total 0
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter3.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter4.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter5.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter6.odf

/home/student/Documents/my_bestseller/editor:
total 0
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter1.odf
-rw-rw-r--. 1 student student 0 Jan 31 19:18 mystery_chapter2.odf

/home/student/Documents/project_plans:
total 4
-rw-rw-r--. 1 student student 0 Jan 31 18:20 season1_project_plan.odf
-rw-rw-r--. 2 student student 0 Jan 31 19:05 season2_project_plan.odf
```

19. 从 serverb 退出。

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

评估

在 workstation 上，运行 **lab files-review grade** 脚本来确认是否成功完成本实验。

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

完成

在 workstation 上，运行 **lab files-review finish** 脚本来结束本实验。此脚本将删除实验练习过程中在 serverb 上创建的所有文件和目录。

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

本实验到此结束。

总结

在本章中，您学到了：

- Linux 系统中的文件被组织到一个颠倒的目录树中，称为文件系统层次结构。
- 绝对路径以 a / 开头并指定文件在文件系统层次结构中的位置。
- 相对路径不以 a / 开头并指定文件相对于当前工作目录的位置。
- 有五个关键命令可用于管理文件：**mkdir**、**rmdir**、**cp**、**mv** 和 **rm**。
- 硬链接和软链接是让多个文件名指向同一数据的不同方式。
- Bash shell 提供了模式匹配、扩展和替换功能，可帮助您高效地运行命令。

章 4

在红帽企业 LINUX 中获取帮助

目标

通过利用本地帮助系统来解决问题。

培训目标

- 在本地 Linux 系统手册页中查找信息。
- 从 GNU Info 中的本地文档查找信息。

章节

- 阅读 man page (及引导式练习)
- 阅读 Info 文档 (及引导式练习)

实验

在红帽企业 Linux 中获取帮助

阅读 MAN PAGE

培训目标

学完本节后，您将能够在本地 Linux 系统手册页中找到相关信息。

MAN 命令简介

本地系统上通常可用的一个文档源是系统手册页，或称为 man page。这些手册页是作为文档所涉及的相应软件包的一部分而提供的，并可使用 **man** 命令从命令行进行访问。

man page 源自过去的 Linux 程序员手册，该手册篇幅很长，足以打印成多个章节。每个章节中都包含了有关特定主题的信息。

Linux 手册的常见章节

章节	内容类型
1	用户命令（可执行命令和 shell 程序）
2	系统调用（从用户空间调用的内核例程）
3	库函数（由程序库提供）
4	特殊文件（如设备文件）
5	文件格式（用于许多配置文件和结构）
6	游戏（过去的有趣程序章节）
7	惯例、标准和其他（协议、文件系统）
8	系统管理和特权命令（维护任务）
9	Linux 内核 API（内核调用）

为区分不同章节中相同的主题名称，man page 参考中在主题后附上了章节编号（用括号括起）。例如，**passwd(1)** 介绍更改密码的命令，而 **passwd(5)** 说明用于存储本地用户帐户的 **/etc/passwd** 文件格式。

若要读取具体的 man page，请使用 **man topic**。内容一次显示一个屏幕。**man** 命令按照字母数字顺序搜索手册章节。例如，**man passwd** 默认情况下显示 **passwd(1)**。要显示要具体章节的 man page 主题，请附上章节编号参数：**man 5 passwd** 显示 **passwd(5)**。

MAN PAGE 导航与搜索

能够高效搜索主题并在 man 手册中导航是一项重要的管理技能。GUI 工具可以让配置常见系统资源变得简单，但使用命令行界面仍然更为高效。要有效地在命令行中导航，您必须能够在 man page 中找到所需的信息。

下表列出了在查看 man page 时的基本导航命令：

Man Page 导航

命令	结果
空格键	向前（向下）滚动一个屏幕
PageDown	向前（向下）滚动一个屏幕
PageUp	向后（向上）滚动一个屏幕
向下箭头键	向前（向下）滚动一行
向上箭头键	向后（向上）滚动一行
D	向前（向下）滚动半个屏幕
U	向后（向上）滚动半个屏幕
/string	在 man page 中向前（向下）搜索 string
N	在 man page 中重复之前的向前（向下）搜索
Shift+N	在 man page 中重复之前的向后（向上）搜索
G	转到 man page 的开头。
Shift+G	转到 man page 的末尾。
Q	退出 man ，并返回到命令 shell 提示符



重要

在执行搜索时，字符串允许使用正则表达式语法。尽管简单文本（例如 **passwd**）可以起到预期作用，但正则表达式使用元字符（例如 \$、*、. 和 ^）可获得更复杂的模式匹配。因此，搜索包含程序表达式元字符的字符串（如 **make \$\$\$**）可能会产生意外的结果。

有关正则表达式和语法的介绍，请参见红帽系统管理二，以及 **regex(7)** man 主题。

阅读 man page

每个主题划分为几个部分。大多数主题共享相同的标题，并以相同的顺序显示。通常，主题不包含所有标题，因为并非所有标题都适用于所有主题。

常见的标题有：

标题

标题	描述
NAME	主题名称。通常是命令或文件名。非常简短的描述。
SYNOPSIS	命令语法的概要。

标题	描述
DESCRIPTION	提供对主题的基本理解的深度描述。
OPTIONS	命令执行选项的说明。
EXAMPLES	有关如何使用命令、功能或文件的示例。
FILES	与 man page 相关的文件和目录的列表。
SEE ALSO	相关的信息，通常是其他 man page 主题。
BUGS	软件中的已知错误。
AUTHOR	有关参与编写该主题的人员的信息。

根据关键字搜索 MAN PAGE

通过 **man -k keyword** 对 man page 执行关键字搜索，这会显示与关键字匹配的 man page 主题和章节编号的列表。

```
[student@desktopX ~]$ man -k passwd
checkPasswdAccess (3) - query the SELinux policy database in the kernel.
chpasswd (8)           - update passwords in batch mode
ckpasswd (8)           - nnrpd password authenticator
fgetpwent_r (3)        - get passwd file entry reentrantly
getpwent_r (3)         - get passwd file entry reentrantly
...
passwd (1)             - update user's authentication tokens
sslpasswd (1ssl)       - compute password hashes
passwd (5)              - password file
passwd.nntp (5)        - Passwords for connecting to remote NNTP servers
passwd2des (3)          - RFS password encryption
...
```

常见的系统管理主题在第 1 节（用户命令）、第 5 节（文件格式）和第 8 节（管理命令）中。运用特定故障排除工具的管理员也使用第 2 节（系统调用）。剩余的章节通常供程序员参考或用于高级管理。



注意

关键字搜索依赖于 **mandb(8)** 命令生成的索引，该命令必须以 **root** 身份运行。该命令每天通过 **cron.daily** 运行，或者通过 **anacrontab** 在启动一小时内运行（如果过期）。



重要

man 命令 **-K**（大写）选项执行全文本页面搜索，而不像 **-k** 选项那仅搜索主题和描述。全文本搜索使用更多系统资源，耗费更长的时间。



参考文献

`man(1)`、`mandb(8)`、`man-pages(7)`、`less(1)`、`intro(1)`、`intro(2)`、`intro(5)`、`intro(7)`、`intro(8)` man page

► 指导练习

阅读 MAN PAGE

在本练习中，您将练习使用 **man** 选项和参数查找相关的信息。

成果

您应能够使用 **man** Linux 手册系统，并通过搜索和浏览查找有用的信息。

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 help-manual start** 命令。它会创建一个名为 **manual** 的文件。

```
[student@workstation ~]$ lab help-manual start
```

- 1. 在 **workstation** 上查看 **gedit** man page。从命令行使用 **gedit** 查看用于编辑特定文件的选项。

使用 **gedit** man page 中的一个选项，通过 **gedit** 打开 **/home/student/manual** 文件，光标位于文件的末尾。

1.1. 查看 **gedit** man page。

```
[student@workstation ~]$ man gedit
```

```
GEDIT(1)      General Commands Manual      GEDIT(1)
NAME
       gedit - text editor for the GNOME Desktop

SYNOPSIS
       gedit [OPTION...] [FILE...] [+LINE[:COLUMN]]
       gedit [OPTION...] -
...output omitted...
```

1.2. 在 **gedit** man page 中，了解用于从命令行编辑特定文件的选项。

```
...output omitted...
       FILE  Specifies the file to open when gedit starts.
...output omitted...
       +LINE  For the first file, go to the line specified by LINE (do not insert
a space between the "+" sign and the number). If LINE is missing, go to the last
line.
...output omitted...
```

按 **q** 键退出 man page。

1.3. 使用 **gedit +** 命令打开 **manual** 文件。+ 选项旁边缺少行号会打开一个作为参数传递的文件，而光标位于最后一行的末尾。

```
[student@workstation ~]$ gedit + manual
```

```
the quick brown fox just came over to greet the lazy poodle!
```

确认文件已打开，且光标位于文件最后一行的末尾。按 **Ctrl+q** 关闭应用程序。

► 2. 阅读 **su(1)** man page。

请注意，如果省略了用户，**su** 命令会假定用户是 **root**。如果 **su** 命令后跟一个短划线 (-)，它将启动一个子登录 shell。如果不使用单短划线，则会创建与用户当前环境匹配的非登录子 shell。

```
[student@workstation ~]$ man 1 su
```

```
SU(1) User Commands SU(1)
NAME
  su - run a command with substitute user and group ID

SYNOPSIS
  su [options] [-] [user [argument...]]

DESCRIPTION
  su allows to run commands with a substitute user and group ID.
  When called without arguments, su defaults to running an interactive
  shell as root.
  ...output omitted...
OPTIONS
  ...output omitted...
  -, -l, --login
    Start the shell as a login shell with an environment similar to a real login
  ...output omitted...
```



注意

注意在一列中列出逗号分隔的选项，如 **-**、**-l** 和 **--login**，它们都会造成相同的行为。

按 **q** 键退出 man page。

► 3. **man** 命令也有自己的 man page。

```
[student@workstation ~]$ man man
MAN(1) Manual pager utils

NAME
  man - an interface to the on-line reference manuals
  ...output omitted...
```

DESCRIPTION

```
man is the system's manual pager. Each page argument given to man is
normally the name of a program, utility or function. The manual page
associated with each of these arguments is then found and displayed.
A section, if provided, will direct man to look only in that section
of the manual.
...output omitted...
```

按 **q** 键退出 man page。

- 4. 所有 man page 都位于 **/usr/share/man**。使用 **whereis** 命令，查找位于 **/usr/share/man** 目录中的二进制文件、源代码和 man page。

```
[student@workstation ~]$ whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1.gz /usr/share/
man/man5/passwd.5.gz
```

- 5. 使用 **man -k zip** 命令来列出关于 ZIP 存档的详细信息。

```
[student@workstation ~]$ man -k zip
...output omitted...
zipinfo (1)           - list detailed information about a ZIP archive
zipnote (1)           - write the comments in zipfile to stdout, edit comments and
    rename files in zipfile
zipsplit (1)          - split a zipfile into smaller zipfiles
```

- 6. 使用 **man -k boot** 来列出包含可在启动时传输到内核的参数列表的 man page。

```
[student@workstation ~]$ man -k boot
...output omitted...
bootctl (1)           - Control the firmware and boot manager settings
bootparam (7)          - introduction to boot time parameters of the Linux kernel
bootup (7)             - System bootup process
...output omitted...
```

- 7. 使用 **man -k ext4** 来查找用于调整 ext4 文件系统参数的命令。

```
[student@workstation ~]$ man -k ext4
...output omitted...
resize2fs (8)          - ext2/ext3/ext4 file system resizer
tune2fs (8)            - adjust tunable filesystem parameters on ext2/ext3/ext4
filesystems
```

完成

在 workstation 上，运行 **lab help-manual finish** 脚本来完成本练习。

```
[student@workstation ~]$ lab help-manual finish
```

本引导式练习到此结束。

阅读 INFO 文档

培训目标

学完本节后，学员应该能够从 GNU Info 中的本地文档查找信息。

GNU INFO 简介

man page 的格式作为命令参考时很有用，但作为普通文档却用处不大。对于此类文档，GNU 项目开发了一种不同的在线文档系统，称为 GNU Info。Info 文档是 Red Hat Enterprise Linux 系统上重要的资源，因为 coreutils 软件包和 glibc 标准库等许多基本的组件和实用程序或是由 GNU 项目开发而成，或是利用了 Info 文档系统。



重要

您可能想知道为什么会有两个本地文档系统，即 man page 和 Info 文档。其中有些原因是基于实用性的考虑，还有一些原因则与多年来各种开源社区开发 Linux 及其应用的方式有关。

man page 的格式更为正式，内容通常记录的是软件包中的特定命令或功能，而且采用独立文本文件的结构。Info 文档则通常涵盖整个特定的软件包，其内容往往是有关如何使用软件的更为实用的示例，而且采用了超文本文档结构。

您应该熟悉这两个系统，以便最大限度利用系统中可用的信息。

阅读 Info 文档

要启动 Info 文档查看器，可使用 **pinfo** 命令。**pinfo** 会在顶级目录中打开。

The screenshot shows a terminal window titled "File: dir Node: Top This is the top of the INFO tree". The text displayed is:

```
This (the Directory node) gives a menu of major topics.  
Typing "q" exits, "?" lists all Info commands, "d" returns here,  
"h" gives a primer for first-timers,  
"mEmacs<Return>" visits the Emacs topic, etc.  
  
In Emacs, you can click mouse button 2 on a menu item or cross reference  
to select it.  
  
* Menu:  
  
Archiving  
* Cpio: (cpio). Copy-in-copy-out archiver to tape or disk.  
* Tar: (tar). Making tape (or disk) archives.  
  
Basics  
* Common options: (coreutils)Common options.  
    Common options.  
* Coreutils: (coreutils). Core GNU (file, text, shell) utilities.  
* Date input formats: (coreutils)Date input formats.  
* File permissions: (coreutils)File permissions.  
    Access modes.  
* Finding files: (find). Operating on files matching certain criteria.  
* ed: (ed). The GNU Line Editor.  
  
Viewing line 25/2002, 1%
```

图 4.1: pinfo Info 文档查看器，顶级目录