命令行快速入门
通过终端控制你的计算机

Zed A. Shaw

Dec 2011

Contents

译者前言

这本书和 Zed 别的书籍一样，强调的是在使用中学习。本书是零基础入门，讲的是最基本的命令行知识。如果你读了《笨办法学 Python》或者《笨方法学 Ruby》，发现里边讲到的命令行你看不懂，那么通过这本小书入门就是合适的。

基本的命令行很简单，这本小书也是练习多于讲解。欢迎反馈翻译中的错误，你可以反馈到 https://github.com/gastlygem/clic cn，同时你也可以在那里获得更新的 PDF 文件。

前言

笔者写这本小书的目的是引导学生上手笔者的其他编程书籍。很多学生不知道命令行界面的基本用法，这阻碍了他们的学习过程。这本书的设计可以让读者一天到一周内就读完，并且学到足够的命令行知识来下手别的书籍。

这本书讲的不是关于系统管理的高深技术，只是为了让菜鸟上手而已。

Chapter 1

准备工作

这本书将带领你做三件事情：

1. 在你的 shell，或者命令行 (command line)，或者终端 (Terminal)，或者 PowerShell 写一些东西。
2. 弄懂你刚写的东西。
3. 自己再多写一些东西。

第一节练习中，你的目的是打开你的终端并确认它能正常工作，以便继续学习下去。

1.1 任务

准备好你的 Terminal，shell，PowerShell，设置好以便快速访问它们。

1.1.1 Mac OSX

对于 Mac OSX 你要做的如下：

1. 按住 COMMAND 键再敲击空格键。
2. 右上方会跳出蓝色的“搜索栏”。
3. 输入：terminal
4. 点击长得像一个黑盒子的 Terminal 应用程序。
5. 这样 Terminal 就打开了。
6. 你可以跑到 Dock 里面按着 CTRL 点击，在打开的菜单中选择 `Options->Keep` 在 Dock 中。

这样你就打开了 Terminal，而且 Dock 里也有了快速访问链接。

1.1.2 Linux

如果你已经在使用 Linux，那我就可以假设你已经知道如何找到 terminal 了。在你的 Window Manager 里搜寻名字像"Shell" 或者"Terminal" 的东西就可以了。

1.1.3 Windows

Windows 下我们将使用 PowerShell。人们以前用的一个叫 cmd.exe 的程序，不过和 PowerShell 比起来它的可用性差很多。如果你用的是 Windows 7 以上的系统，就照下面的做：

1. 点击 Windows 菜单。
2. 在搜索框中输入：powershell
3. 敲击回车键。

如果你装的不是 Windows 7，那你应该认真考虑一下升级事宜。如果你实在不想升级，那就试着从微软的下载中心安装一下。这得靠你自己了，因为我没有装 Windows XP，写不出安装流程来，不过希望 XP 下的 PowerShell 的使用体验也是一样的吧。

1.2 知识点

你学会了如何打开你的 terminal，这是继续本书所必须的。

Note 1 *躲开所谓的黑客和他们的 zsh*

如果你有一个挺聪明而且懂 Linux 的朋友，再如果他介绍 bash 之外的 shell 给你，那你应该忽略他们的建议。我教你的是 bash，就是这样。他们会说 zsh 会让你的 IQ 多长 30 个点，并且让你在股票市场上赚到腰缠万贯，别理他们就是了。你的目的只是学会足够的技能，在你这个技能等级上，使用哪个 shell 其实不会影响到什么东西。

接下来的一个建议就是躲开 IRC 以及那些“黑客”常去的地方。他们会教你一些破坏你电脑的命令并以此为乐。例如这条经典的 `rm -rf /`，千万别输这条命令！ 离黑客远点，如果你需要帮助，就找你能信任的人，别去找网上乱七八糟的二货。

1.3 更多任务

本节练习有一个很大的“更多任务”部分。其他的练习没这么多的额外任务要做，不过我要让你通过记忆的方式向自己灌输足够的知识。跟着我走就行了，这会让你后面的学习像丝一般顺畅。

1.3.1 Linux/Mac OSX

用索引卡片写下所有的命令，一张卡片写一条，正面写下命令，背面写下解释。每次看书就学习一次，每次学个 15 分钟左右。

pwd print working directory（打印工作目录）

hostname my computer's network name（电脑在网络中的名称）

mkdir make directory（创建路径）

cd change directory（更改路径）

ls list directory（列出路径下的内容）

rmdir remove directory（删除路径）

pushd push directory（推入路径）

popd pop directory（推出路径）

cp copy a file or directory（复制文件或路径）

mv move a file or directory（移动文件或路径）

less page through a file（逐页浏览文件）

cat print the whole file（打印输出整个文件）

xargs execute arguments（执行参数）

find find files（寻找文件）

grep find things inside files（在文件中查找内容）

man read a manual page（阅读手册）

apropos find what man page is appropriate（寻找恰当的手册页面）

env look at your environment（查看你的环境）

echo print some arguments（打印一些参数）

export export/set a new environment variable（导出/设定一个新的环境变量）

exit exit the shell（离开 shell）

sudo DANGER! become super user root DANGER!（成为超级用户或 root，危险命令！）

chmod change permission modifiers（修改文件许可权限）

chown change ownership（修改文件的所有者）

1.3.2 Windows

If you're using Windows then here's your list of commands:

pwd print working directory（打印工作目录）

hostname my computer's network name（电脑在网络中的名称）

mkdir make directory（创建路径）

cd change directory（更改路径）

ls list directory（列出路径下的内容）

rmdir remove directory（删除路径）

pushd push directory（推入路径）

popd pop directory（推出路径）

cp copy a file or directory（复制文件或路径）

robocopy robust copy（更可靠的复制命令）

mv move a file or directory（移动文件或路径）

more page through a file（打印输出整个文件）

type print the whole file（打印输出整个文件）

forfiles run a command on lots of files（在一大堆文件上面运行一条命令）

dir /r find files（寻找文件）

select-string find things inside files（在文件中查找内容）

help read a manual page（阅读手册）

helpctr find what man page is appropriate（寻找恰当的手册页面）

echo print some arguments（打印一些参数）

set export/set a new environment variable（导出/设定一个新的环境变量）

exit exit the shell（离开 shell）

runas DANGER! become super user root DANGER!（成为超级用户或 root，危险命令!）

attrib change permission modifiers（修改文件许可权限）

iCACLS change ownership（修改文件的所有者）

不停地练习，直到你可以看到一条命令，然后能立即说出它的功能为止。然后反过来也能说出实现每个功能所需的命令。通过这样的方式你可以为自己建立语汇，不过如果你觉得烦了，也别强迫自己在上面花太多时间。

Chapter 2

路径, 文件夹, 目录 (pwd)

2.1 任务

接下来我要教你如何阅读我展示给你的这些终端会话。你不需要将所有的东西都键入终端，只是其中的一部分内容而已：

1. 你不需要键入 `$` (unix) 或者 `>` (Windows)。这个只是为了标记这是一个终端会话而已。
2. 你写完 `$ or >` 后面的内容后需要敲击回车键。所以如果你看到了 `$ pwd`，那你需要输入 `pwd` 再敲击一次回车键。
3. 你可以看到输出的内容前面也有一个 `$` 或者 `>` 提示。这些是输出内容，你的输出内容和我的应该是一样的。

让我们先试一个命令，看看你有没有弄明白：

Linux/Mac OSX 练习 2

```
$ pwd
/Users/zedshaw
$
```

Windows 练习 2

```
PS C:\Users\zed> pwd

Path
----
C:\Users\zed

PS C:\Users\zed>
```

Note 2 命令提示符之 *Windows vs Unix*

为了节约空间同时让你集中精力到重要的命令细节上面，本书将把命令行一开始的部分（例如上面的 `PS C:\Users\zed` ）省略掉，只留下一个小小的 `>` 部分。这意味着你的命令行和这里看到的会有一点不同，不过这个是正常的，你无须操心。

记住从现在开始，我只通过 `>` 来告诉你这是一个命令行提示。

对于 Unix 命令行提示也一样，不过 Unix 有点不一样，人们习惯使用 `$` 来表示命令提示符。

2.2 知识点

你的命令行和我的看上去不一样，你的可能在 `$` 前面显示了你的用户名以及计算机名。Windows 下看上去也会不一样，不过关键的基本格式都是这样的：

1. 有一个命令提示符。
2. 你输入一条命令，例如这里的 pwd。
3. 你得到一些输出。
4. 重复上述步骤。

你正好还学会了 *`pwd`* 的功能，它的意思是“打印工作目录 (print working directory)”。目录是什么东西？就是文件夹而已。文件夹和目录是一样的东西，这两个词可以交替使用。当你打开文件浏览器，通过图形界面寻找文件，那你就是在访问文件夹。这些文件夹和我们后面要用到的目录完全是同一回事。

2.3 更多任务

1. 输入 *`pwd`* 20 次，每次输入都念一遍“print working directory”。
2. 记下命令行显示的路径，用你的文件浏览器找到这个位置。
3. 我不是开玩笑。写 20 遍，并且朗读出来。别抱怨了，照我说的做。

Chapter 3

计算机名 (hostname)

3.1 任务

Linux/Mac OSX 练习 3

```
$ hostname
Zeds-MacBook-Pro.local
$
```

Windows 练习 3

```
> hostname
zed-PC
>
```

3.2 知识点

这就是你的计算机的名字，或者至少是它的名字之一吧。你的计算机名字和我的很可能不一样，是什么都有可能。

3.3 更多任务

1. 和上一习题一样，输入这条命令 20 次，同时朗读出来。

Part I

Directories

Chapter 4

创建目录 (mkdir)

4.1 任务

Linux/Mac OSX 练习 4

```
$ mkdir temp
$ mkdir temp/stuff
$ mkdir temp/stuff/things
$ mkdir -p temp/stuff/things/frank/joe/alex/john
$
```

Windows 练习 4

```
> mkdir temp


    Directory: C:\Users\zed


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:02 AM            temp


> mkdir temp/stuff


    Directory: C:\Users\zed\temp
```

```
Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:02 AM            stuff


> mkdir temp/stuff/things


    Directory: C:\Users\zed\temp\stuff

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            things


> mkdir temp/stuff/things/frank/joe/alex/john


    Directory: C:\Users\zed\temp\stuff\things\frank\joe\alex

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            john


>
```

4.2 知识点

现在我们输入了一条以上的命令。这些命令是使用 mkdir 的不同方法。mkdir 的功能是什么呢？它是用来 make directory（创建目录）的。你不该问这个问题吧？你应该已经通过索引卡记住这些了才对。如果你不知道这条，那就说明你要继续在索引卡上下功夫。

创建目录是什么意思？目录又可以叫作“文件夹”，它们是同样的东西。你上面所做的是在逐层深入地创建目录，目录有时又叫做路径，这里相当于是说“先到 temp，再到 stuff，然后到 things，这就是我要到的地方。”这是对计算机发出的一系列指向，告诉计算机你想要把某个东西放到计算机硬盘的某个文件夹（路径）里。

Note 3 *Windows,* 斜杠和反斜杠

本书中我使用斜杠 / (slash) 来表示路径，因为所有的计算机都是这么做的。不过，Windows 用户应该知道，反斜杠 \ (backslash) 也可以实现同样的功能，别的 Windows 用户可能认为这才是期望的用法。

4.3 更多任务

1. 你可能觉得路径的概念还是有些绕人。别担心，我们会做大量的练习让你深入理解。
2. 在 temp 下面再创建 20 个别的目录，深度可以各不相同，然后用文件浏览器检查你创建的目录。
3. 创建一个名称包含空格的目录，方法是为名称添加一个引号：`mkdir "I Have Fun"`
4. 如果 **`temp`** 目录已经存在，你要创建它时将会得到一条错误信息。使用 *`cd`* 转到一个别的目录下面试试创建 temp 目录，如果是 Windows 的话，**`Desktop`** 是个不错的选择。

Chapter 5

更改目录 (cd)

5.1 任务

我将再教一遍终端会话的方法：

1. `$` (unix) 和 `>` (Windows) 是不需要写出来的。
2. 你写完 `$ or >` 后面的内容后需要敲击回车键。如果你看到 `$ cd temp`，那你需要输入的是 `cd temp` 然后敲回车。
3. 敲回车后你会看到输出，输出的前面也会有一个 `$` 或者 `>` 提示符。

Linux/Mac OSX 练习 5

```
$ cd temp
$ pwd
~/temp
$ cd stuff
$ pwd
~/temp/stuff
$ cd things
$ pwd
~/temp/stuff/things
$ cd frank/
$ pwd
~/temp/stuff/things/frank
$ cd joe/
$ pwd
~/temp/stuff/things/frank/joe
$ cd alex/
$ pwd
~/temp/stuff/things/frank/joe/alex
$ cd john/
```

```
$ pwd
~/temp/stuff/things/frank/joe/alex/john
$ cd ..
$ cd ..
$ pwd
~/temp/stuff/things/frank/joe
$ cd ..
$ cd ..
$ pwd
~/temp/stuff/things
$ cd ../../..
$ pwd
~/
$ cd temp/stuff/things/frank/joe/alex/john
$ pwd
~/temp/stuff/things/frank/joe/alex/john
$ cd ../../../../../../../../
$ pwd
~/
$
```

Windows 练习 5

```
> cd temp
> pwd

Path
----
C:\Users\zed\temp


> cd stuff
> pwd

Path
----
C:\Users\zed\temp\stuff


> cd things
> pwd

Path
```

```
----
C:\Users\zed\temp\stuff\things


> cd frank
> pwd

Path
----
C:\Users\zed\temp\stuff\things\frank


> cd joe
> pwd

Path
----
C:\Users\zed\temp\stuff\things\frank\joe


> cd alex
> pwd

Path
----
C:\Users\zed\temp\stuff\things\frank\joe\alex


> cd john
> pwd

Path
----
C:\Users\zed\temp\stuff\things\frank\joe\alex\john


> cd ..
> cd ..
> cd ..
> pwd

Path
----
C:\Users\zed\temp\stuff\things\frank
```

```
> cd ../..
> pwd

Path
----
C:\Users\zed\temp\stuff


> cd ..
> cd ..
> cd temp/stuff/things/frank/joe/alex/john
> cd ../../../../../../../
> pwd

Path
----
C:\Users\zed


>
```

5.2 知识点

你在上一节习题中创建了不少的目录，现在你所做的是通过 *cd* 命令在它们之间往来。在上面的终端会话中，我通过使用 *pwd* 来检查自己的当前路径，所以记住别把 *pwd* 的输出也当作要键入的东西。例如第三行有一条 `~/temp` 但这只是 *pwd* 的输出而已。*别把它也输入了。*

你应该还看到了我可以使用 `..` 来移动到目录的上一层。

5.3 更多任务

在图形界面计算机上面学习使用命令行界面（command line interface，CLI），很重要的一部分就是弄明白命令行和图形界面是如何互相配合工作的。我开始使用计算机的时候，GUI 还不存在，所有的事情都要通过 DOS 命令窗口（CLI）来完成。后来计算机越变越强大，人人都能用到图形界面了。对我来说，命令行的目录和图形界面的文件夹都很容易理解。

然而现在大部分人都不理解命令行界面以及路径和目录这些概念。其实这些东西也很难学会，只有不停地学习和使用 CLI，直到有一天你豁然开朗，所有 GUI 下做的事情都和 CLI 下的事情对应起来了。

早日理解的方法是花一些时间来通过图形界面的文件浏览器来寻找目录，然后通过命令行去访问它们，这就是你接下来要做的练习：

1. 用一条命令 cd 到 `joe` 目录。
2. 用一条命令回到 `temp` 目录，不过不要退到太远了。
3. 找出如何用一条命令 cd 到你的“home 目录”
4. cd 到你的 Documents 目录，然后通过你的图形文件浏览器找到这个目录。(Finder, Windows 浏览器, 等等).
5. cd 到你的 Downloads 目录，然后通过文件浏览器找到这个位置。
6. 用文件浏览器找到另外一个位置，然后 cd 到这个位置。
7. 记得你可以为包含空格的目录家一个引号吧？对于任何命令你都可以这么做。加入你创建了一个叫 `I Have Fun` 的文件夹，那你就可以使用 `cd "I Have Fun"` 这条命令。

Chapter 6

列出目录下的内容 (ls)

6.1 任务

开始之前，确认你已经 *cd* 到了 temp 的上一级目录。如果你不确定现在在哪个目录里，那就是用 *pwd* 找出来。

Linux/Mac OSX 练习 6

```
$ cd temp
$ ls
stuff
$ cd stuff
$ ls
things
$ cd things
$ ls
frank
$ cd frank
$ ls
joe
$ cd joe
$ ls
alex
$ cd alex
$ ls
$ cd john
$ ls
$ cd ..
$ ls
john
$ cd ../../../
```

```
$ ls
frank
$ cd ../../
$ ls
stuff
$
```

Windows 练习 6

```
> cd temp
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            stuff


> cd stuff
> ls


    Directory: C:\Users\zed\temp\stuff


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            things


> cd things
> ls


    Directory: C:\Users\zed\temp\stuff\things


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            frank
```

```
> cd frank
> ls


    Directory: C:\Users\zed\temp\stuff\things\frank


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            joe


> cd joe
> ls


    Directory: C:\Users\zed\temp\stuff\things\frank\joe


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            alex


> cd alex
> ls


    Directory: C:\Users\zed\temp\stuff\things\frank\joe\alex


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            john


> cd john
> ls
> cd ..
> ls
```

```
    Directory: C:\Users\zed\temp\stuff\things\frank\joe\alex


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            john


> cd ..
> ls


    Directory: C:\Users\zed\temp\stuff\things\frank\joe


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            alex


> cd ../../..
> ls


    Directory: C:\Users\zed\temp\stuff


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            things


> cd ..
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            stuff
```

```
>
```

6.2 知识点

ls 命令列出了你当前所在的目录下的内容。你看到我使用了 *cd* 变更目录，然后列出里边的内容，这样我就知道接下来该到哪个目录下面去了。

ls 命令有很多的选项，不过你后面会通过学习 *help* 来自己找到这些东西。

6.3 更多任务

1. 输入每一条命令！ 要学习这些命令，你必须输入这些命令。光阅读是不够的。这一点我以后就不跟你啰嗦了。
2. 如果你用 Unix，那就在 `temp` 目录中试一下 `ls -lR` 命令。
3. Windows 下一样的功能可以通过 `dir -R` 完成。
4. 使用 *cd* 进到别的目录下，然后通过 *ls* 看看里边有什么内容。
5. 在笔记本里记下你的问题。我知道你会有些问题，因为对于这条命令我并没讲全。
6. 记住如果你在路径中迷失了，就使用 *ls* 和 *pwd* 来找出你的当前路径，然后通过 *cd* 到达你的目的路径即可。

Chapter 7

删除路径 (rmdir)

7.1 任务

Linux/Mac OSX 练习 7

```
$ cd temp
$ ls
stuff
$ cd stuff/things/frank/joe/alex/john/
$ cd ..
$ rmdir john
$ cd ..
$ rmdir alex
$ cd ..
$ ls
joe
$ rmdir joe
$ cd ..
$ ls
frank
$ rmdir frank
$ cd ..
$ ls
things
$ rmdir things
$ cd ..
$ ls
stuff
$ rmdir stuff
$ pwd
```

```
~/temp
$
```

Windows 练习 7

```
> cd temp
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:03 AM            stuff


> cd stuff/things/frank/joe/alex/john/
> cd ..
> rmdir john
> cd ..
> rmdir alex
> cd ..
> rmdir joe
> cd ..
> rmdir frank
> cd ..
> ls


    Directory: C:\Users\zed\temp\stuff


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:14 AM            things


> rmdir things
> cd ..
> ls
```

```
    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/17/2011   9:14 AM            stuff


> rmdir stuff
> pwd

Path
----
C:\Users\zed\temp


> cd ..
>
```

7.2 知识点

我现在将命令混到了一起，所以你要集中精力确认输入完全相同。你的每一个错误都是不够集中精力导致的。如果你发现自己犯了很多的错误，那就休息一会，或者今天就别接着学习了，等明天再鼓足精神继续。

在本例中你将学会如何删除（remove）一个目录。这很容易，你只要到你要移除的目录的上一层，然后输入 `rmdir DIR`，将 DIR 替换成你要删除的目录的名称即可。

7.3 更多任务

1. 再创建 20 个目录，然后把它们都删除掉 M。
2. 创建一个逐层嵌套的目录，一共嵌套 10 层，然后进到目录中，将这些目录逐层删掉，就更我上面做的一样。
3. 如果你要移除的目录中包含一些内容，那么你就会碰到一个错误信息。后面的练习中我会告诉你如何移除这样的目录。

Chapter 8

在多个目录中切换 (pushd, popd)

8.1 任务

Linux/Mac OSX 练习 8

```
$ cd temp
$ mkdir -p i/like/icecream
$ pushd i/like/icecream
~/temp/i/like/icecream ~/temp
$ popd
~/temp
$ pwd
~/temp
$ pushd i/like
~/temp/i/like ~/temp
$ pwd
~/temp/i/like
$ pushd icecream
~/temp/i/like/icecream ~/temp/i/like ~/temp
$ pwd
~/temp/i/like/icecream
$ popd
~/temp/i/like ~/temp
$ pwd
~/temp/i/like
$ popd
~/temp
$ pushd i/like/icecream
~/temp/i/like/icecream ~/temp
$ pushd
```

```
~/temp ~/temp/i/like/icecream
$ pwd
~/temp
$ pushd
~/temp/i/like/icecream ~/temp
$ pwd
~/temp/i/like/icecream
$
```

Windows 练习 8

```
> cd temp
> mkdir -p i/like/icecream


    Directory: C:\Users\zed\temp\i\like


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/20/2011  11:05 AM            icecream


> pushd i/like/icecream
> popd
> pwd

Path
----
C:\Users\zed\temp


> pushd i/like
> pwd

Path
----
C:\Users\zed\temp\i\like


> pushd icecream
> pwd
```

```
Path
----
C:\Users\zed\temp\i\like\icecream


> popd
> pwd

Path
----
C:\Users\zed\temp\i\like


> popd
>
```

8.2 知识点

如果你用到这些命令，那你就非常接近编程领域了，不过这些命令非常好用，所以我非教你不可。这些命令可以让你临时跑到某个不同的目录中，然后再回到你之前的目录，并且方便地在两者之间切换。

pushd 命令会将你目前所在的目录“推送 (push)”到一个列表中以供后续使用，然后让你转入到另一个目录中。它的意思大致是：“记住我的现在位置，然后到这个地方去。”

popd 命令会将你上次 push 过的目录从列表中“弹出 (pop)”，然后让你回到这个被“弹出”的目录。

最后，在 Unix 中的 *pushd* 有点不同，如果你运行时不添加任何参数，那么它就会让你再当前目录和你上一次 push 过的目录之间切换，这个方法可以让你很方便地在两个目录之间切换。不过 *PowerShell* 中这样做是不灵的。

8.3 更多任务

1. 使用这些命令在你计算机目录之间多切换几次。
2. 删掉 `i/like/icecream` 这些目录，然后自己创建一些目录，在它们之间切换。
3. 向自己解释 *pushd* 和 *popd* 的输出的意义。有没有发现它的工作模式有点像一个堆栈。
4. 前面已经教过了，记住 `mkdir -p` 会创建一个完整的多层目录，即使中间目录不存在也能成功。这也是我创建本习题一开始所做的事情。

Part II

Files

Chapter 9

创建空文件 (Touch, New-Item)

9.1 任务

Linux/Mac OSX 练习 9

```
$ cd temp
$ touch iamcool.txt
$ ls
iamcool.txt
$
```

Windows 练习 9

```
> cd temp
> New-Item iamcool.txt -type file
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---        12/17/2011   9:03 AM            iamcool.txt


>
```

9.2 知识点

你学会了如何创建空文件。在 Unix 中你要用 *touch*，它还有一个功能就是修改文件的时间。我除了用它创建空文件以外，很少用它做别的事情。Windows 中没有这个命令，所以你要学习使用 *New-Item* 命令，它实现的功能是一样的，不过它还可以创建新目录。

9.3 更多任务

1. Unix：创建一个目录，转到这个目录下，然后在其中创建一个文件，然后回到上一层目录，对你创建的目录运行 *rmdir*，你应该会看到一个错误，试着弄懂为什么你会碰到这个错误。
2. Windows：做同样的事情，不过你不会看到错误。你会看到一个提示，问你是否真的要删除这个目录。

Chapter 10

复制文件 (cp)

10.1 任务

Linux/Mac OSX 练习 10

```
$ cd temp
$ cp iamcool.txt neat.txt
$ ls
iamcool.txt        neat.txt
$ cp neat.txt awesome.txt
$ ls
awesome.txt        iamcool.txt        neat.txt
$ cp awesome.txt thefourthfile.txt
$ ls
awesome.txt                iamcool.txt                neat.txt                thefourthfile.txt
$ mkdir something
$ cp awesome.txt something/
$ ls
awesome.txt                iamcool.txt                neat.txt                something                thef
$ ls something/
awesome.txt
$ cp -r something newplace
$ ls newplace/
awesome.txt
$
```

Windows 练习 10

```
> cd temp
```

```
> cp iamcool.txt neat.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt


> cp neat.txt awesome.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 awesome.txt
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt


> cp awesome.txt thefourthfile.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 awesome.txt
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt


> mkdir something
```

```
    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            something


> cp awesome.txt something/
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 awesome.txt
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt


> ls something


    Directory: C:\Users\zed\temp\something


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 awesome.txt


> cp -recurse something newplace
> ls newplace


    Directory: C:\Users\zed\temp\newplace


Mode                 LastWriteTime     Length Name
```

```
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 awesome.txt


>
```

10.2 知识点

现在你学会了复制（copy）文件。很简单，就是把一个文件拷贝成一个新文件而已。在这个练习中我还创建了一个新目录，然后将文件拷贝到其中去。

我现在要告诉你一个关于程序员和系统管理员的秘密：他们可懒了。我是懒人，我的朋友们也是懒人，这也是我们用计算机的原因。我们让计算机为我们做各种无聊的事情。你学到现在，所做的事情就是重复输入各种无趣的命令，并通过这个过程学会这些命令，但实际工作中不是这样子的。在实际工作中，如果你发现某个任务需要通过无趣的重复工作来完成，那么很可能已经有程序员找出让这个任务变得更简单的方法了，只不过你不知道而已。

另外一件要告诉你的，就是程序员其实没有你想象的那么聪明。如果你过度思考要输入的命令的名称，结果很可能是过而不及。取而代之的，你应该去想这个命令的名字，然后直接试试这个名称或者类似这个名称的缩写。如果还是不灵，那就问问别人，或者上网搜索。不过碰到 ROBOCOPY 这么二的命令名字那就谁也没辙了。

10.3 更多任务

1. 练习使用 *cp -r* 命令去复制一些包含文件的目录。
2. 将一个文件拷贝到你的 home 目录或者桌面。
3. 从图形界面找到你拷贝过的文件，然后用文本编辑器打开它们。
4. 有没有发现我有时会在目录的结尾放一个斜杠 (slash) `/` ？这样做的目的是保证输入的名称确实是一个目录，所以如果这个目录不存在，那么我就会看到一个错误信息。

Chapter 11

移动文件 (mv)

11.1 任务

Linux/Mac OSX 练习 11

```
$ cd temp
$ mv awesome.txt uncool.txt
$ ls
newplace        uncool.txt
$ mv newplace oldplace
$ ls
oldplace        uncool.txt
$ mv oldplace newplace
$ ls
newplace        uncool.txt
$
```

Windows 练习 11

```
> cd temp
> mv awesome.txt uncool.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
```

```
d----         12/22/2011   4:52 PM            newplace
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt
-a---         12/22/2011   4:49 PM          0 uncool.txt


> mv newplace oldplace
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            oldplace
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt
-a---         12/22/2011   4:49 PM          0 uncool.txt


> mv oldplace newplace
> ls newplace


    Directory: C:\Users\zed\temp\newplace


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         12/22/2011   4:49 PM          0 awesome.txt


> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
```

```
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            newplace
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt
-a---         12/22/2011   4:49 PM          0 uncool.txt


>
```

11.2 知识点

移动 (move) 文件，或者换种说法，重命名 (rename) 文件。很简单，提供旧名称和新名称即可。

11.3 更多任务

1. 将一个文件从 **newplace** 目录移动到另一个目录，然后将它移动回来。

Chapter 12

查看文件内容 (less, MORE)

要完成这个练习，你将用到你学过的一些命令，另外你还需要一个文本编辑器来创建纯文本 (.txt) 文件，以下是你要做的准备工作：

1. 打开文本编辑器，在新文本中输入一些东西。在 OSX 中你可以用 TextWrangler，在 Windows 下你可以使用 Notepad++，在 Linux 中你可以用 Gedit，随便什么编辑器都可以。
2. 保存该文件到桌面，将其命名为 **ex12.txt**。
3. 在命令行 (shell) 使用你学过的命令将该文件拷贝 (copy) 到你的工作目录，也就是 **temp** 目录中去。

做好准备工作以后，就可以完成任务了。

12.1 任务

Linux/Mac OSX 练习 12

```
$ less ex12.txt
[displays file here]
$
```

就是这样子。要推出 *less*，只要输入 q 即可。这个 q 的意思是推出 (quit)。

Windows 练习 12

```
> more ex12.txt
[displays file here]
>
```

Note 4 *"displays file here"* 表示略掉了输出内容

上面的输出中我用 `[displays file here]` 来指代程序的输出。在后面的练习中，如果碰到复杂情况无法向你展示输出内容，我就会用这个来指代你的输出。你的屏幕不会显示这句话。

12.2 知识点

这是查看文件内容的一个方法。它有用的地方在于，如果文件内容有很多行，它会将其分页，这样就会每次显示一页。在“更多任务”中你会看到更多相关的练习。

12.3 更多任务

1. 再次打开你的文本文件，重复复制粘贴若干次，让你的文本长度约等于 50 至 100 行。
2. 将它再次复制到 `temp` 目录下，这样你就可以通过命令行查看了。
3. 再做一遍练习，不过这次你要逐页浏览文档。在 Unix 下使用空格键和 `w` 键上下翻页，使用方向键也可以，不过在 Windows 下你就只能用空格键向下逐页浏览了。
4. 查看你创建的空文件的内容。
5. *`cp`* 命令会覆盖已经存在的文件，复制文件时要留意这一点。

Chapter 13

流水式文件内容显示 (cat)

你需要更多的准备工作，这个过程也会让你习惯这个工作流程：你在一个程序中创建文件，然后通过命令行对其进行处理。使用练习 12 中的 文本编辑器 创建一个叫 **ex13.txt** 的文件，不过这次将其直接保存到 **temp** 目录下。

13.1 任务

Linux/Mac OSX 练习 13

```
$ less ex13.txt
[displays file here]
$ cat ex13.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
$ cat ex12.txt
Hi there this is cool.
$
```

Windows 练习 13

```
> more ex13.txt
[displays file here]
> cat ex13.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
```

```
> cat ex12.txt
Hi there this is cool.
>
```

记住，我写的 `[displays file here]` 是表示我略掉了命令的输出，这样我就不用把东西详尽地展示给你了。

13.2 知识点

我的诗怎么样？拿个诺贝尔奖没问题吧？不管怎样，你已经学到了第一个命令，而我只是让你检查你的文件已经在那里了。然后你使用 *cat* 将文件内容显示到屏幕上。这个命令会将整个文件一次输出到屏幕，不会分页也不会中间停顿。为了展示这一点，我让你对 **ex12.txt** 执行 *cat* 命令，结果就是一下输出了文本中所有的行。

13.3 更多任务

1. 再创建几个文本文件，然后用 *cat* 逐一打开。
2. Unix：试试 `cat ex12.txt ex13.txt`，看看结果是怎样的。
3. Windows：试试 `cat ex12.txt,ex13.txt` 看看结果。

Chapter 14

删除文件 (rm)

14.1 任务

Linux/Mac OSX 练习 14

```
$ cd temp
$ ls
uncool.txt                iamcool.txt                neat.txt                something                thefo
$ rm uncool.txt
$ ls
iamcool.txt                neat.txt                something                thefourthfile.txt
$ rm iamcool.txt neat.txt thefourthfile.txt
$ ls
something
$ cp -r something newplace
$
$ rm something/awesome.txt
$ rmdir something
$ rm -rf newplace
$ ls
$
```

Windows 练习 14

```
> cd temp
> ls


    Directory: C:\Users\zed\temp
```

```
Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            newplace
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt
-a---         12/22/2011   4:49 PM          0 uncool.txt


> rm uncool.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            newplace
d----         12/22/2011   4:52 PM            something
-a---         12/22/2011   4:49 PM          0 iamcool.txt
-a---         12/22/2011   4:49 PM          0 neat.txt
-a---         12/22/2011   4:49 PM          0 thefourthfile.txt


> rm iamcool.txt
> rm neat.txt
> rm thefourthfile.txt
> ls


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         12/22/2011   4:52 PM            newplace
d----         12/22/2011   4:52 PM            something


> cp -r something newplace
```

```
> rm something/awesome.txt
> rmdir something
> rm -r newplace
> ls
>
```

14.2 知识点

这里我们将上一个练习中的文件清理掉了。先前我让你用 *rmdir* 来删除包含文件的目录，但是你的操作失败了，失败的原因是你不能用这条命令来删除包含文件的目录。要移除这样的目录，你需要先删除文件，或者循环删除目录下的所有内容，这也是本练习的结尾所做的事情。

14.3 更多任务

1. 删除掉 `temp` 目录下至今为止的所有内容。
2. 在笔记本上记下来：循环移除文件时要小心操作。

Chapter 15

管道和重定向

15.1 任务

Linux/Mac OSX 练习 15

```
$ cat ex12.txt ex13.txt | less
$ cat < ex13.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
$ less < ex12.txt
$ less < ex12.txt | cat | less
$ cat ex13.txt > ex15.txt
$ cat ex15.txt
I am a fun guy.
Don't you know why?
Because I make poems,
that make babies cry.
$
```

Windows 练习 15

```
> cd ..
> cd temp
> cat ex12.txt,ex13.txt | more
> echo "I am a new file." > ex15.txt
> cat ex15.txt
I am a new file.
```

```
> cat ex15.txt > another.txt
> cat another.txt
I am a new file.
> cat ex15.txt | more
>
```

15.2 知识点

现在我们将接触命令行比较酷的知识点：重定向 (redirection)。这个概念的意义在于你可以修改程序的输入和输出的走向，你要通过 < (小于号)、> (大于号)、| (管道符) 来做到这些。以下是详细解说：

| | 将左边命令的输出导向到右边命令中去。第 1 行向你演示了这一点。

< < 将右边的文件作为输入发送给左边的程序。你看到第 2 行所做的就是这个。*PowerShell 不支持这种操作*。

> > 将左边命令的输出写入到右边的文件中去。第 9 行展示了这一点。

>> >> 将左边命令的输出*追加 (append)*到右边的文件中去。我在第 9 行做了这样的操作。

还有一些别的符号，不过现在我们只学这些就够用了。

15.3 更多任务

1. 制作一些索引卡用来辅助记忆这些符号。将符号写在一侧，它的意义写在另一侧，像记忆命令一样将这些符号记下来。

Chapter 16

通配符匹配

16.1 任务

Linux/Mac OSX 练习 16

```
$ cd temp
$ ls *.txt
ex12.txt        ex13.txt        ex14.txt        uncool.txt
$ ls ex*.*
ex12.txt        ex13.txt        ex14.txt
$ ls e*
ex12.txt        ex13.txt        ex14.txt
$ ls *t
ex12.txt        ex13.txt        ex14.txt        uncool.txt
$ cat *.txt > bigfile.txt
$ rm *.txt
$ ls
$
```

Windows 练习 16

```
> cd temp
> ls *.txt


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
```

```
----                 -------------     ------ ----
-a---        12/22/2011   5:23 PM         38 another.txt
-a---        12/22/2011   5:23 PM         38 ex15.txt


> ls ex*.*


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---        12/22/2011   5:23 PM         38 ex15.txt


> ls e*


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---        12/22/2011   5:23 PM         38 ex15.txt


> ls *t


    Directory: C:\Users\zed\temp


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---        12/22/2011   5:23 PM         38 another.txt
-a---        12/22/2011   5:23 PM         38 ex15.txt


> cat *.txt
I am a new file.
I am a new file.
> rm *.txt
> ls
```

```
>
```

16.2 知识点

有时你需要用一条命令处理一批文件。处理的方法是使用星号 * 来表示“任何文件”。每当你使用了星号，命令行将会对匹配到非星号字符的文件生成一个列表。

这个练习中你用了各种方法列出你生成的文件。我的目录里还有一些多余的文件，你也许也有一些。关键点在于，当你写下 `*.txt` 时，你的意思是说“任何以.txt 结尾的文件。”

最后我们用 `rm *.txt` 来删除 `temp` 目录下的所有.txt 文件。

16.3 更多任务

1. 将 * 加到你的速记卡中。在背后写下“通配符，用来匹配任何内容，例如 *.txt”。

Part III

Searching

Chapter 17

寻找文件 (find, DIR -R)

17.1 任务

这个练习将三个概念合并到一条命令中来。我将展示给你如何找到你所有的文本文件，并主页浏览它们。

Linux/Mac OSX 练习 17

```
$ cd temp
$ find . -name "*.txt" -print
$ cd ..
$ find . -name "*.txt" -print | less
$ cd ..
$ find . -name "*.txt" -print | less
$
```

Windows 练习 17

```
> dir -r
> cd ..
> dir -r -filter "*.txt"
> cd ..
> dir -r | more
>
```

17.2 知识点

你学会了用 *find* (Windows 的 *dir -r*) 命令来搜索所有以 **.txt** 结尾的文件，然后用 *less* (Windows 的 *more*) 来查看结果。

以下是 Unix 下的详解：

1. 首先我进到 `temp` 目录下。
2. 然后我简单执行了寻找 (find) 命令，因为这里也没有多少 `.txt` 文件。这个命令跟一句话差不多：“嘿，*find*，从这里 (.) 开始，找到所有叫做”*.txt” 的文件，然后将它们打印出来。”将命令用跟计算机对话的方式记下来，这是一个不错的记忆方法。
3. 接下来我到上一层目录执行了相同的命令，不过这次我将输出转向到了 *less* 中。这条命令执行起来可能会花一些时间，需要你耐心等待一下。
4. 然后我又执行了一遍，不过这次的时间可能会真的有点长，所以你可以随时用 CTRL-c 把它中断掉。

Windows 下也差不多：

1. 然后我简单执行了 DIR 命令，因为这里也没有多少 `.txt` 文件。这条命令的意思是“列出当前目录以及所有子目录中的所有东西”。这是一个循环命令 (recursive command)。
2. 接下来我到上一层目录执行了同样的命令，不过这次我把输出转向到 *more*。这条命令时间会有点长，需要耐心等待。
3. 然后我又执行了一遍，不过这次的时间可能会真的有点长，所以你可以随时用 CTRL-c 把它中断掉。

17.3 更多任务

1. Unix：找出你的 *find* 索引卡，在描述面写下“find STARTDIR -name WILDCARD -print”。记住这个命令的格式，用我给你的句式辅助记忆。
2. Windows：同上，不过要写的是“dir -r -filter”。
3. 你可以将 . 替换为任何目录，试着换一个目录搜索一下。
4. 寻早你计算机中所有的视频文件，从 home 目录开始找起，将输出用 `>` 保存到一个文件中。记住你可以用 `SOMECOMMAND > SOMEFILE.txt` 这样的格式，它会将 SOMECOMMAND 的输出写入到 SOMEFILE.txt 中。

Chapter 18

文件查找内容 (grep, select-string)

18.1 任务

我来教你一个快速将文本写入一个文件的方法。如果你执行 `cat > somefile.txt`，*cat* 将会把你接下来键入的所有东西都写入这个文件中，Windows 下你需要执行 `echo > somefile.txt`。很重要的一点是，你需要用 CTRL-d 来结束你的输入。

如果你现在学不会这个操作，你可以使用文本编辑器来创建 **newfile.txt** 和 **oldfile.txt**。

Linux/Mac OSX 练习 18

```
$ cd temp
$ cat > newfile.txt
This is a new file.
This is a new file.
This is a new file.
$ cat > oldfile.txt
This is a old file.
This is a old file.
This is a old file.
$ grep new *.txt
newfile.txt:This is a new file.
newfile.txt:This is a new file.
newfile.txt:This is a new file.
$ grep old *.txt
oldfile.txt:This is a old file.
oldfile.txt:This is a old file.
oldfile.txt:This is a old file.
$ grep file *.txt
newfile.txt:This is a new file.
newfile.txt:This is a new file.
```

```
newfile.txt:This is a new file.
oldfile.txt:This is a old file.
oldfile.txt:This is a old file.
oldfile.txt:This is a old file.
$
```

这个技巧在 Windows 下的用法差不多，只不过你要执行的是 `echo > somefile.txt`，而且接下来命令行会提示让你输入每一行，当你输入一个空行（直接回车）时，这条命令也就结束了。

Windows 练习 18

```
> cd temp
> echo > newfile.txt

cmdlet Write-Output at command pipeline position 1
Supply values for the following parameters:
InputObject[0]: This is a new file.
InputObject[1]: This is a new file.
InputObject[2]: This is a new file.
InputObject[3]:
> echo > oldfile.txt

cmdlet Write-Output at command pipeline position 1
Supply values for the following parameters:
InputObject[0]: This is a old file.
InputObject[1]: This is a old file.
InputObject[2]: This is a old file.
InputObject[3]:
> select-string new *.txt

newfile.txt:1:This is a new file.
newfile.txt:2:This is a new file.
newfile.txt:3:This is a new file.


> select-string old *.txt

oldfile.txt:1:This is a old file.
oldfile.txt:2:This is a old file.
oldfile.txt:3:This is a old file.


> select-string file *.txt
```

```
newfile.txt:1:This is a new file.
newfile.txt:2:This is a new file.
newfile.txt:3:This is a new file.
oldfile.txt:1:This is a old file.
oldfile.txt:2:This is a old file.
oldfile.txt:3:This is a old file.


>
```

18.2 知识点

你创建了两个几乎一样的文件，只不过其中一个里边有“new”，而另一个是“old”。然后你在这些文件中搜索不同的文字，你可以查找单词，不过如果你为要寻找的东西加上引号，你也可以寻找整句内容。

18.3 更多任务

1. 使用引号寻找”new file” 和”old file” 以及”This is”。
2. 在你创建的视频文件列表（或者任何文件列表）中查找你想要找到的某个文件。
3. Unix：你可以使用 `-i` 让 *grep* 忽略大小写。执行 `grep -i new *.txt` 试试看。

Part IV

Help

Chapter 19

命令行的帮助 (man, HELP)

19.1 任务

Linux/Mac OSX 练习 19

```
$ man find
$ man less
$ man man
$ man grep
$
```

Windows 练习 19

```
> help dir
> help select-string
> help help
> help cp
>
```

19.2 知识点

你可以在 Unix 下使用 ***man*** 命令，在 Windows 下使用 ***help*** 命令来查找某条命令的相关信息。一路走来我算是向你使了个阴招，其实我一开始就可以告诉你让你查找帮主，然后去阅读每条命令的功能，不过假如我这么做的话，你还是会晕头，因为你还没有掌握一些基本的命令，不知道目录的原理，也不懂通配符是什么，如此这般你也看不明白这些帮助。

我希望你能原谅我直到现在才交给你这个有用的工具，不过从此以后，如果你忘了某条命令的功能，那你就使用帮助命令即可。

19.3 更多任务

1. 使用 *man* 或者 *help* 来阅读你记忆过的每一条命令的帮助文档。

Chapter 20

寻找帮助 (apropos, HELP)

20.1 任务

Linux/Mac OSX 练习 20

```
$ apropos search
$ apropos find
$ apropos remove
$ apropos directory
$
```

Windows 练习 20

```
> help *Al*
> help *Dir*
> help *Find*
> help *remove*
```

20.2 知识点

有时你会忘记某个命令是怎么写的，但你记得这条命令的功能，这个命令会在所有命令的帮助文件中查找，然后把可能相关的命令名称汇报给你。

实话讲，我只在别无选择时才会用到这个命令。一般我会先上网搜索，通常我能搜索到更好的命令说明，如果我依然找不到命令的名称，我会在命令列表中一一浏览下去，直到找出我要的命令为止。

20.3 更多

1. 一样地，使用这里的命令去寻找你的列表中的所有命令。

Part V

Sessions

Chapter 21

环境变量 (env, echo, Env:)

21.1 任务

Linux/Mac OSX 练习 21

```
$ env
TERM_PROGRAM=Apple_Terminal
TERM=xterm
SHELL=/bin/bash
OLDPWD=/Users/zed/temp
USER=zed
COMMAND_MODE=unix2003
PATH=/opt/local/bin:/opt/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin
PWD=/Users/zed/
LANG=en_US.UTF-8
PS1=$
SHLVL=1
HOME=/Users/zed
LOGNAME=zed
_=/usr/bin/env
$ env | grep zed
OLDPWD=/Users/zed//temp
USER=zed
PWD=/Users/zed/
HOME=/Users/zed
LOGNAME=zed
$ echo $USER
zed
$ echo $PWD
/Users/zed/
```

```
$ export TESTING="1 2 3"
$ echo $TESTING
1 2 3
$ env | grep TESTING
TESTING=1 2 3
$
```

Windows 练习 *21*

```
> get-childitem Env:

Name                           Value
----                           -----
ALLUSERSPROFILE                C:\ProgramData
APPDATA                        C:\Users\zed\AppData\Roaming
...


> $env:PROMPT
$P$G
> $env:TEMP
C:\Users\zed\AppData\Local\Temp
> $env:OS
Windows_NT
>
```

21.2 知识点

你的命令行 (shell) 有一些“隐藏的变量”，它们可以改变程序执行的效果。这个练习中我首先打印出了我的环境变量，将它们输出到屏幕，然后我又执行了一遍，将输出转向到 *grep* 并找到了包含我的用户名的环境变量。最后，我将一个叫做 TESTING 的环境变量的值设为”1 2 3”。

这个你也许不会经常用到，不过有时你要配置某些东西时会要修改这些环境变量。PATH 变量就是一个很好的例子，它决定了系统是按怎样的顺序搜索到你要执行的命令的。后面的练习中你将学到如何修改 PATH。

21.3 更多任务

1. 我要求你上网搜索，研究一下如何修改你的计算机中的 PATH。试着只用命令行来完成这个任务。

Chapter 22

修改环境变量 (export, Env:)

22.1 任务

Linux/Mac OSX 练习 22

```
$ export TESTING="bada bada bing"
$ echo $TESTING
bada bada bing
$ unset TESTING
$ echo $TESTING

$ env | grep TESTING
$
```

Windows 练习 22

```
> get-childitem Env:

Name                           Value
----                           -----
APPDATA                        C:\Users\zed\AppData\Roaming
COMPUTERNAME                   ZED-PC
...


> $env:TESTING = "bada bada bing"
> $env:TESTING
bada bada bing
> remove-item Env:\TESTING
```

```
> get-childitem Env: | select-string TESTING
>
```

22.2 知识点

你可以修改环境变量的值，还可以取消环境变量的设置，这样它就不存在于你的环境中了。你可以通过将环境变量设为空（""）来删除它，这需要通过命令来实现。

22.3 更多任务

1. 列出你的所有环境变量，然后上网搜索它们的意义。
2. 再阅读一遍关于 env 的 man 帮助档，这条命令还能实现些什么功能？

Chapter 23

离开命令行 (exit)

23.1 任务

Linux/Mac OSX 练习 23

```
$ exit
```

Windows 练习 23

```
> exit
```

23.2 知识点

你的最后一个练习是如何离开命令行终端。这个是在没什么好讲的，不过我会让你做更多事情。

23.3 更多任务

作为你最后的练习，我要求你使用帮助系统来查看一些命令，你需要自己研究并学会使用这些命令。

Unix 的命令列表：

1. xargs
2. sudo
3. chmod
4. chown

Windows 的命令列表：

1. forfiles
2. runas
3. attrib
4. icacls

学习这些命令的功能和用法，练习使用它们，然后将它们加到你的索引卡中。

Part VI

Epilogue

Chapter 24

将来的路

你已经完成了这本快速入门书。到这里你的水平基本上达到了能使用 shell 的程度了。其实要学的技巧和命令用法还有很多，这里我会再给你一些最后的阅读和研究方向。

24.1 Unix Bash 参考资料

你一直在用的 shell 叫做 Bash。它不见得是最牛的 shell，不过你在到处都能看到它，而且它也有不少的功能，所以作为入门是很不错的。接下来我向你提供一些关于 Bash 的链接，你应该好好阅读一下：

Bash Cheat Sheet http://cli.learncodethehardway.org/bash_cheat_sheet.pdf，作者 Raphael，CC license。

Reference Manual http://www.gnu.org/software/bash/manual/bashref.html

24.2 PowerShell 参考资料

Windows 下能用的也就只有 PowerShell 了。以下是关于 PowerShell 的一些有用链接：

Owner's Manual http://technet.microsoft.com/en-us/library/ee221100.aspx

Cheat Sheet http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=7097

Master Powershell http://powershell.com/cs/blogs/ebook/default.aspx

24.3 向前进

从现在开始你应该不再害怕使用命令行了。如果你立志要做一名程序员，本书算是一个最好的起点，你可以领会到计算机其实就是一个“语言机器”，而 shell 则类似于一个很迷你很易学的编程语言。

要想成为命令行高手，我的建议是你要强迫自己每天都使用它，不管用起来有多难。比较困难的一点是在没有视觉提示的前提下要记住一些命令。图形界面的好处在于你可以通过图形提示得到一些线索，从而知道工具该怎么使用。而命令行界面下，你就不得不从零开始挖掘命令的功能，一开始你会觉得这件事情有些恼人的。

不过我有一个技巧可以减少你的痛苦。为自己创建一个小抄，写下你经常用到的命令行技巧。使用命令行去做事情，

如果你碰到一个会多次用到的命令，那就把它写下来。下次你要用这个命令时，看一下你的小抄，这样你就会慢慢记住这些命令了。

最终你就用不到这样的小抄了。其实对我来说每天用到的命令也不过只有十来个，大部分已经包含在这本书里边了。要记住十来个命令不是什么难事，所以没有什么能阻挡你的。

如果你遇到困难，你可以通过 help@learncodethehardway.org 联系我，我会帮你解决。