Linux
典藏大系

Linux Shell

命令行及脚本编程实例详解

刘艳涛　编著

清华大学出版社
北　京

内 容 简 介

本书理论结合实践，全面、系统地介绍了 Linux Shell（Bash）脚本编程的语法、命令、技巧等内容。本书偏重于实践教学，在讲解理论知识时，通过一些典型实例让读者了解理论知识在实际环境中的应用，并对易混淆和较难理解的知识点做了重点分析，以加深读者对知识的理解。**另外，作者专门为本书录制了高清配套教学视频，以帮助读者高效学习，同时也提供了本书实例源程序以方便读者学习。**

本书共 15 章，分为两篇。主要内容包括：Linux 及 Linux Shell 简介、初识 Linux Shell、常用 Shell（Bash）命令、Shell 命令进阶、Shell 编程基础、Shell 的条件执行、Bash 循环、Shell 函数、正则表达式、脚本输入处理、Shell 重定向、管道和过滤器、捕获、sed 和 awk，以及其他 Linux Shell 种类介绍。

本书使用了大量的实例详细地介绍了 Bash 的语法及各种技巧，并以循序渐进的方式讲解了 Linux Shell（Bash）的各种特性，让读者能够迅速上手，并能学以致用。对于初次接触 Linux Shell 的读者，本书是一本很好的自学教材；对于接触过 Linux Shell 的读者，本书可以作为进阶读物或随时查阅的技术手册；另外，本书也可以作为高等学校相关专业的教材和各类培训学校的教材。

图书在版编目（CIP）数据

Linux Shell 命令行及脚本编程实例详解 / 刘艳涛编著. —北京：清华大学出版社，2015
（Linux 典藏大系）
ISBN 978-7-302-37862-4

Ⅰ. ①L…　Ⅱ. ①刘…　Ⅲ. ①Linux 操作系统 – 程序设计　Ⅳ. ①TP316.89

中国版本图书馆 CIP 数据核字（2014）第 202651 号

责任编辑： 夏兆彦
封面设计： 欧振旭
责任校对： 胡伟民
责任印制：

出版发行： 清华大学出版社
网　　址： http://www.tup.com.cn, http://www.wqbook.com
地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 26.5　　**字　　数：** 662 千字
版　　次： 2015 年 1 月第 1 版　　**印　　次：** 2015 年 1 月第 1 次印刷
印　　数： 1～ 000
定　　价：　　元

产品编号：059488-01

前　言

在当今的互联网世界，想必最为流行的一个词就是“云计算”了，而且云计算又引领了大数据时代的到来。而 Linux 在推动云计算方面起到了举足轻重的作用。比如，当今最著名的商业云计算平台“亚马逊弹性计算云（EC2）”就是基于 Linux 的。这就需要在 Linux 服务器上进行大量的数据处理和管理工作，以及一些应用的部署和监测，这时就需要命令行和 Shell 脚本的帮助。在 Linux 系统中，我们通常是在命令行下完成一些管理和配置的任务，并通过 Shell 脚本将一些重复或定期的任务自动化，通过短短几行脚本自动地将大部分手头工作搞定，从而节省大量的时间。而且理解 Shell 脚本也可以让你更好地了解操作系统。Shell 脚本还可以和许多外部命令行工具结合起来完成信息查询、文本处理、任务定时自动化以及监测系统之类的工作。当然，伴随着这些便利性的还有巨大的破坏性。比如，稍不留神，你可能就会将整个根目录全部毁掉，或者错误地处理重要的配置文件。这时，了解 Linux 命令行和 Shell 脚本相关的细节、遵循 Linux 使用规范就显得尤其重要了。

本书面向系统管理员，基于 Linux 系统的软件开发和测试人员，以及所有想有效使用 Linux 系统的爱好者。书中系统而全面地介绍了 Shell（Bash）脚本编程的语法、命令和技巧等内容，结合大量的实例进行讲解，你可以将其作为参考，或是作为自己编写脚本时的灵感源泉。

关于“Linux 典藏大系”

“Linux 典藏大系”是清华大学出版社自 2010 年 1 月以来陆续推出的一个图书系列，截止 2013 年，已经出版了 10 余个品种。该系列图书涵盖了 Linux 技术的方方面面，可以满足各个层次和各个领域的读者学习 Linux 技术的需求。该系列图书自出版以来获得了广大读者的好评，已经成为了 Linux 图书市场上最耀眼的明星品牌之一。其销量在同类图书中也名列前茅，其中一些图书还获得了“51CTO 读书频道”颁发的“最受读者喜爱的原创 IT 技术图书奖”。该系列图书在出版过程中也得到了国内 Linux 领域最知名的技术社区 ChinaUnix（简称 CU）的大力支持和帮助，读者在 CU 社区中就图书的内容与活跃在 CU 社区中的 Linux 技术爱好者进行广泛交流，取得了良好的学习效果。

本书特色

1. 视频讲解

为了帮助读者更加高效、直观地学习，编著者为本书重点内容专门录制了配套教学视

频，需读者自行下载。

2. 内容全面

本书将理论与实践相结合，全面介绍了 Linux 系统的常用命令及 Shell 脚本编程所需的知识点。书中对 Linux Shell 脚本编程的基本概念、语法、命令、技巧和较难理解的知识点都配合图示和实例进行了详细讲解。

3. 循序渐进

本书从最基本的 Linux Shell 命令开始讲解，逐步深入到 Linux Shell 脚本编程，让读者可以迅速掌握 Linux Shell 的各种特性，并能在实际开发中加以使用。

4. 实例丰富

本书偏重于实践教学，书中的每一个理论知识都给出了具体的典型实例。例如，对每一个 Linux 常用命令、Linux Shell 的相关概念及 Shell 脚本编程的相关知识等，都列举了大量实例供读者了解这些知识点在实际环境中的应用。

5. 经验传授

本书是基于编著者多年的 Linux 系统管理和 Linux 平台程序设计的经验总结而来，书中给出了大量的经验和技巧，尽力消除读者在学习 Linux Shell 编程时可能会遇到的各种障碍，从而更加迅速而高效地掌握 Shell 脚本编程。

本书内容及体系结构

第1篇　Linux Shell基础和使用（第1～4章）

本篇介绍 Linux 命令行和 Linux Shell 的基础知识，包括 Linux 及 Linux Shell 简介、Bash 简介、Bash 启动和退出脚本、Shell 中的变量、Shell 中的扩展、创建和使用别名，以及 Shell 下的常用命令等。这些内容都是学习后续章节所必须要掌握的基础知识。

第2篇　Shell 脚本编程（第5～15章）

本篇主要介绍 Shell 脚本编程所需的知识，包括 Shell 脚本中的注释、Shell 变量进阶、Shell 的算术运算、如何退出脚本、如何调试脚本、Shell 脚本编程风格、Shell 的条件执行、Shell 中的循环和控制语句、Shell 函数、正则表达式、脚本的输入处理、Shell 重定向、管道和过滤器、Shell 中的捕获、sed 和 awk，以及其他 Shell 中的介绍等内容。这些内容几乎涵盖了日常使用 Shell 脚本编程的方方面面。

本书读者对象

- Linux Shell 编程入门新手；
- Linux Shell 编程进阶人员；

- 广大 Linux 程序设计人员；
- Linux 系统管理员；
- 网站管理工程师；
- Linux 培训机构的学员；
- Linux Shell 编程爱好者；
- 需要一本案头必备查询手册的人员。

本书配套资源获取方式

本书涉及的源程序和配套教学视频等学习资料需要读者自行下载。请到清华大学出版社的网站（www.tup.com.cn）上搜索到本书页面，按提示下载。也可到本书服务网站 www.wanjuanchina.net 上的相关版块下载。

关于编著者

本书由刘艳涛主笔编写。其他参与编写的人员有陈冠军、陈浩、黄振东、蒋庆学、李代叙、李世民、李思清、李云龙、李志刚、刘存勇、刘燕珍、龙哲、吕轶、牟春梅、屈明环、石峰、史艳艳、宋宁宁、王德亮、王俊清、王雅宁、翁盛鑫。

您在阅读本书的过程中若有疑问，请发 E-mail 和我们联系。E-mail 地址：bookservice2008@163.com。

编著者

目　　录

第 1 篇　Linux Shell 基础和使用

第 1 篇　Linux Shell 基础和使用

第 1 章　Linux 及 Linux Shell 简介

欢迎来到 Linux Shell 的世界，在我们开始真正的 Linux Shell 之旅前，先让我们简单地了解一下关于 Linux 和 Linux Shell 的历史及其一些相关的基本概念，以便为我们接下来的学习打下一个较好的基础。希望你通过本章的学习，能对 Linux Shell 有一个初步的了解。

1.1　关于 Linux

1.1.1　什么是 Linux

Linux 是自由开源的类 Unix 操作系统。该操作系统的内核由莱纳斯·托瓦兹在 1991 年 10 月 5 日首次发布。

严格来讲，术语 Linux 只表示操作系统的内核本身，但通常采用“Linux 内核”来表达该意思。Linux 则常用来指基于 Linux 内核的完整操作系统，包括 GUI 组件和许多其他实用工具。

Linux 最初是作为支持 Intel x86 架构的个人计算机的一个自由操作系统开发的，目前 Linux 已经被移植到更多的计算机硬件平台。世界上 500 个最快的超级计算机 90%以上运行 Linux 发行版或变种，包括最快的前 10 名超级计算机运行的都是基于 Linux 内核的操作系统。Linux 也被广泛应用在嵌入式系统上，如手机、平板电脑、路由器、电视和电子游戏机等。在移动设备上广泛使用的 Android 操作系统就是基于 Linux 内核的。

Linux 的发展是自由软件和开源软件联盟的最著名的例子之一。只要遵循 GNU 通用公共许可证，任何个人和机构都可以使用、修改和发布 Linux 的底层源代码。通常情况下，Linux 被打包成供桌面应用和服务器应用的 Linux 发行版。一些主流的 Linux 发行版包括 Debian（及其派生版本，例如 Ubuntu 和 Linux Mint）、Red Hat Enterprise Linux（及其派生版本，例如 Fedora 和 CentOS）、openSUSE（及其商业版 SUSE Linux Enterprise Server），还有 Arch Linux。Linux 发行版包含 Linux 内核、配套的实用程序和库，通常还有满足发行版使用目的的大量应用软件。

通常情况下，面向桌面应用的 Linux 发行版包括 X Windows 系统和一个相应的桌面环境，例如 GNOME 或 KDE。一些这样的发行版会包含一个用于较老的或低性能计算机的较少资源集中的桌面，例如 LXDE 或 Xfce。一个用于服务器应用的发行版可能会从标准安装中略去所有的图形环境，而包含其他的一些软件，比如，Apache HTTP 服务和一个 SSH 服务。因为 Linux 是一个自由软件，所以任何人都可以创建一个符合自己应用需求的发行版。

1.1.2　谁创建了 Linux

1991 年，莱纳斯·托瓦兹开始了一个项目，它之后成为了 Linux 内核。它最初是托瓦兹用于访问大学里的 UNIX 服务器的一个终端模拟器。他专门为他当时正在使用的硬件写了一个独立于操作系统的程序，因为他想使用他的 80386 处理器的新计算机的功能。这个程序的开发是在使用 GNU C 编译器的 MINIX 操作系统上完成的，即 Linux 的前身。

如托瓦兹在他的《Just for Fun》书中所写，他最终意识到他编写了一个操作系统内核。1991 年 8 月 25 日他在 Usenet 上发布了这个系统。

1.1.3　Linux 在日常生活中的使用

你可以把 Linux 作为一个服务器操作系统使用，或作为一个你个人计算机上的独立操作系统使用。作为一个服务器操作系统，它为客户端提供不同的服务和网络资源。一个服务器操作系统必须具有以下特性：

- 稳定的；
- 强壮的；
- 安全的；
- 高性能的。

Linux 提供以上所有特性，并且它是自由和开源的。它作为一个杰出的操作系统可以应用于：

- 台式计算机；
- 网站服务器；
- 软件开发工作站；
- 网络监控工作站；
- 工作组服务器；
- 杀手级网络服务，例如 DHCP、防火墙、路由、FTP、SSH、邮件、代理以及代理缓存服务器等等。

1.1.4　Linux Kernel 是什么

如前面所说，Linux 内核，即 Linux 操作系统的核心。它主要由以下模块组成：

- 进程管理；
- 定时器；
- 中断管理；
- 内存管理；
- 模块管理；
- 虚拟文件系统接口；
- 文件系统；
- 设备驱动程序；

- 进程间通信；
- 网络管理；
- 系统引导。

Linux 内核决定了谁将使用这些资源，可以使用多长时间，以及什么时候可以使用这些资源。它在计算机硬件和各种应用程序之间起到了媒介的作用，如图 1.1 所示。

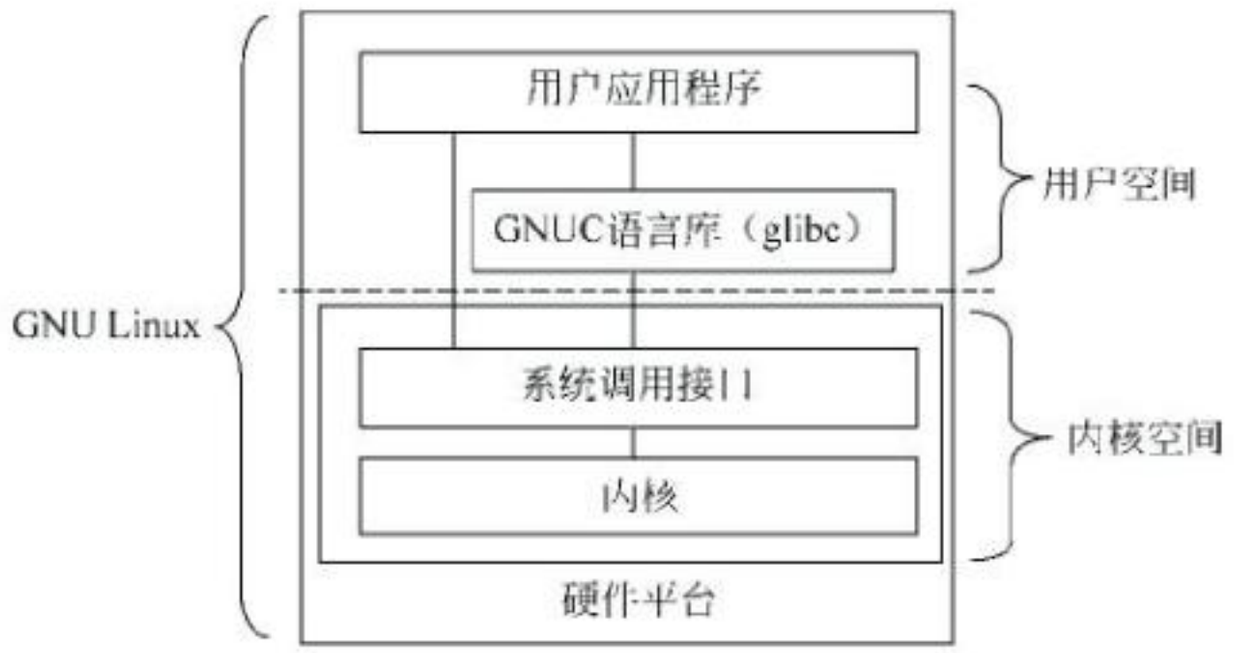

图 1.1　Linux 内涵

1.1.5　Linux 的理念

如之前所述，Linux 是类 Unix 的操作系统，Unix 的理念是一套基于 Unix 操作系统顶级开发者们的经验提出的软件开发的准则和哲学。因此这些理念也同样适用于 Linux 操作系统。

- 小即是美；
- 让程序只做好一件事；
- 可移植性比效率更重要；
- 一切即文件——使用方便而且把硬件作为文件处理是安全的；
- 使用 Shell 脚本来提高效率和可移植性；
- 避免使用可定制性低下的用户界面；
- 所有程序都是数据的过滤器。

1.2　什么是 Linux Shell

Linux Shell 是用户和 Linux 内核之间的接口程序，为用户提供使用操作系统的接口。当从 Shell 向 Linux 传递命令时，内核会做出相应的反应。

- Shell 是一个用户程序，或是一个为用户与系统交互提供的环境。
- 它是一个执行从标准输入设备（比如键盘或文件）读入的命令的语言解释程序，它拥有自己内建的 Shell 命令集，Shell 也能被系统中其他应用程序所调用。
- 当你登录或打开控制台时 Shell 就会运行。
- Shell 不是系统内核的一部分，但是它使用系统内核执行程序、创建文件等。

我们可以通过多种方式来访问和使用 Shell：

- 终端—— Linux 桌面提供基于 GUI 的登录系统。一旦登录你就可以通过运行 X 终端（XTerm）、Gnome 终端（GTerm）或 KDE 终端（KTerm）应用程序来访问 Shell。
- 安全 Shell 连接（SSH）——可以通过它远程登录服务器或工作站来访问其 Shell。
- 使用控制台——一些 Linux 系统同样提供基于文本的登录系统。通常情况下，登录系统后就可以直接访问 Shell。

当普通用户成功登录时，系统将执行一个 Shell 程序，Shell 进程会提供一个命令行提示符。作为默认值，普通用户用“$”作提示符，超级用户（root）用“#”作提示符。一旦出现了 Shell 提示符，就可以输入命令名称及命令所需的参数，输入回车后，Shell 将执行这些命令。

在 Shell 执行命令时，Shell 首先检查命令是否是内部命令，若不是再检查是否是一个应用程序（这里的应用程序可以是 Linux 本身的实用程序，如 date 和 cat，也可以是购买的商业程序，如 rtds，或是自由软件，如 emacs），Shell 在搜索路径里寻找这些应用程序（搜索路径是一个存放可执行程序的目录列表）。如果输入的命令不是一个内部命令并且在搜索路径里没有找到这个可执行文件，Shell 将会显示一条错误信息。如果能够成功找到命令，该命令将被分解为系统调用并传给 Linux 内核。

在 Shell 下，你可以使用如下按键组合来编辑和回调命令。

- CTRL+W：删除光标位置前的单词。
- CTRL+U：清空行。
- ↑，↓方向键：查看命令历史。
- Tab：自动补全文件名、目录名和命令等等。
- CTRL + R：搜索先前使用的命令。
- CTRL + C：中止当前命令。
- CTRL + D：退出登录 Shell。
- ESC + T：调换光标前的两个单词。

当用户准备结束登录对话进程时，可以输入 logout 命令、exit 命令或按 CTRL+D 组合键，结束登录。

Linux Shell 的另一个重要特性是它自身就是一个解释型的程序设计语言，Shell 程序设计语言支持绝大多数在高级语言中能见到的程序元素，如函数、变量、数组和程序控制结构等。Shell 编程语言简单易学，任何在提示符中能输入的命令都可以放到一个可执行的 Shell 脚本中。

1.3　Shell 的种类

Linux（以及 Unix 或类 Unix）中的 Shell 有多种类型，其中最常用的种类有 Bourne Shell（sh）、C Shell 和 Korn Shell。这三种 Shell 各有优缺点。

Bourne Shell 是 UNIX 最初使用的 Shell，并且在每种 UNIX 上都可以使用。Bourne Shell 在 Shell 编程方面相当优秀，但是在处理与用户的交互方面不如其他几种 Shell。

Bourne-Again Shell（bash）是 Linux 系统中最常用的 Shell。它是 Bourne Shell 的扩展，

与 Bourne Shell 完全向后兼容，并且在 Bourne Shell 的基础上增加、增强了很多特性，具有很多特色，可以提供如命令补全、命令编辑和命令历史等功能，它还包含了很多 C Shell 和 Korn Shell 中的优点，有灵活和强大的编程接口，同时又有很友好的用户界面。

C Shell（csh）C Shell 是一种比 Bourne Shell 更适于编程的 Shell，它的语法和用法与 C 语言很相似，Linux 为喜欢使用 C Shell 编程的人提供了 TCSH。

TCSH 是与 C Shell 兼容的增强版本。它包括命令行编辑、可编程单词补全、拼写校正、历史命令替换、作业控制和类似 C 语言的语法。

Korn Shell（ksh）集合了 C Shell 和 Bourne Shell 的优点，并和 Bourne Shell 完全兼容。Linux 系统提供了 ksh 的扩展，它支持任务控制，可以在命令行上挂起、后台执行、唤醒或终止程序。

Linux 中还包括了一些其他的 Shell 类型，如比较流行的 ash 和 zsh 等。但无论哪一种 Shell，它最主要的功用都是解译使用者在命令行提示符中输入的指令。在 MS-DOS 中，也有一种 Shell，它的名字是 COMMAND.COM，它也用于同样的工作，只是它显然没有 Linux Shell 这样强大。每种 Shell 都有它的用途及各自的命令语法和提供不同的内建功能。有些 Shell 是有专利的，有些则可从互联网上直接免费获得。

我们可以使用如下命令查看系统中所有可用的 Shell：

```
-bash-3.2$ cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/tcsh
/bin/csh
/bin/ksh
```

我们看到此文件中包含了多行，每行都是一种 Shell，它代表此系统支持多种 Shell。

用户登录到 Linux 系统时，由/etc/passwd 这个文件决定用户将要使用哪种 Shell，比如我们来查看 root 账号在/etc/passwd 这个文件中的定义：

```
-bash-3.2$ grep root /etc/passwd
root:x:0:0:System Admin:/root:/bin/bash
```

我们可以看到在输出结果中，以冒号“:”分隔的最后一个字段就是定义此账号在登录后所使用的 Shell，由此可知此实例中，root 账号所使用的 Shell 是 bash。

我们还可以使用如下命令来查看账号当前使用的 Shell 的类型：

```
-bash-3.2$ echo $SHELL
/bin/bash
```

或是

```
-bash-3.2$ ps -p $$
  PID TTY          TIME CMD
23579 pts/0    00:00:00 bash
```

1.4　怎样使用 Shell

要使用 Shell，你只需简单地输入命令即可，命令即是一个用于执行特定任务的计算机

程序。

如果你的系统启动后进入的是文本模式，那么当你登录系统后就可以直接使用 Shell，你可以在登录后的 Shell 中输入命令并执行。如果你的系统是以图形界面的模式启动的，例如 GNOME 桌面或是 KDE 桌面，那么你可以在图形界面中单击“应用程序->系统工具->终端”来打开一个 Shell。或者，你可以按 Ctrl+Alt+F1 组合键切换到虚拟控制台并使用你的用户名和密码登录。若想切换回图形界面模式，可以简单地按 Alr+F7 组合键。

Linux 终端提供了一个让你简单地与 Shell（例如 bash）交互的手段。Shell 不过是一个解释并执行你在命令行提示符中输入的命令的程序。当启动 GNOME、KDE 或 X Window 终端时，这些应用程序启动你的系统账号中所指定的默认 Shell。你可以随时切换到不同的 Shell。接下来，我们来简单了解一下 Gnome 终端的使用和配置。

Gnome 终端程序是完全可以配置的，你可以通过设置如下选项来定义一些属性：

- 前景和背景色；
- 窗口标题；
- 字体大小和类型；
- 回滚缓冲区等。

1.5　Shell 脚本是什么

Shell 脚本就像早期 dos 年代的.bat，最简单的功能就是将许多指令汇整在一起，让使用者很容易地就能够一个操作执行多个命令，主要是方便管理员进行设置或者管理用的。但是它比 Windows 下的批处理更强大，它提供了数组、循环、条件以及逻辑判断等重要功能，让使用者可以直接以 Shell 来写程序，比用其他编程语言编写的程序效率更高，毕竟它使用了 Linux/Unix 下的命令。

Shell 脚本是利用 Shell 的功能所写的一个程序，这个程序是纯文本文件格式，将一些 Shell 的语法与指令写在里面，然后用正则表达式、管道命令以及数据流重定向等功能，以实现我们所需要的功能。

Shell 脚本是 Linux/Unix 编程环境的基本组成部分。

Shell 脚本一般由以下几部分构成：

- Shell 关键字——例如 if...else，for do...done。
- Shell 命令——例如 export，echo，exit，pwd，return。
- Linux 命令——例如 date，rm，mkdir。
- 文本处理功能——例如 awk，cut，sed，grep。
- 函数——通过函数把一些常用的功能放在一起。例如，/etc/init.d 目录中的大部分或全部系统 Shell 脚本所使用的函数都包含在文件/etc/init.d/functions 中。
- 控制流语句——例如 if...then...else 或执行重复操作的 Shell 循环。

每个 Shell 脚本都有它的用途，例如，备份文件系统和数据库到网络存储服务器。Shell 脚本可以像 Linux 下的一个命令一样被执行。

1.6　为什么使用 Shell 脚本

Shell 脚本的应用知识对于每一个想熟练地管理 Linux 操作系统的人是必须的，即使你可能从来不必写脚本。比方说在 Linux 机器启动时，它执行/etc/rc.d 目录中的 Shell 脚本来加载系统配置和运行服务，那么详细地理解这些启动脚本，对于我们分析系统的行为或是可能修改这些脚本将是很重要的。

学习编写 Shell 脚本并不难，因为它的语法简单易懂，类似于直接调用命令行的功能并串联在一起，并且只有几种规则需要学习。大部分简短的脚本可以第一次就正确执行，即使要调试长的脚本也是简单的。

总的来说，我们使用 Shell 具有如下一些原因：

- 使用简单；
- 节省时间。可以把冗长的重复的一连串命令合并成一条简单的命令；
- 可以创建你自己的自动化工具和应用程序；
- 使系统管理任务自动化；
- 因为脚本经过很好的测试，所以使用脚本做类似配置服务或系统管理任务时，发生错误的机会将大大减少。

我们经常会用到 Shell 脚本的实例有：

- 监控你的 Linux 系统；
- 备份数据和创建快照；
- 创建邮件告警系统；
- 查找耗尽系统资源的进程；
- 查找是否所有的网络服务都正常运行等等。

1.7　实例：创建你的第一个 Shell 脚本

要想成功地写一个 Shell 脚本，你需要做以下三件事情：

- 写一个脚本；
- 允许 Shell 执行它；
- 把它放到 Shell 可以找到的地方。

一个 Shell 脚本就是一个包含 ASCII 文本的文件，因此可以使用一个文本编辑器来创建一个脚本。文本编辑器是一个类似于读写 ASCII 文本文件的文字处理机的程序。有很多种文本编辑器可在 Linux 系统上使用，它们既可以用于命令行环境也可以用于图形界面环境，你可以根据喜好来选择适合你的文本编辑器。

现在，打开你的文本编辑器并编写包含如下内容的第一个 Shell 脚本：

```
#!/bin/bash
#My first script

ls -l .*
```

保存你的文件，在这里，我们就索性将其命名为 my_first。

脚本的第一行是很重要的。它是一个告诉 Shell 使用什么程序解释脚本的特别指示。在本例中使用的是/bin/bash。其他脚本语言比如 Perl、awk、python 等也同样使用这个机制。

脚本的第二行是一个注释。每一行中出现在“#”符号后的任何内容都将被 bash 忽略。一旦你的脚本变得很大且很复杂时，注释将是极其重要的。它们被程序员用于解释代码的用途，以便其他程序员可以弄清楚。

脚本的最后一行是 ls 命令，它将列出当前目录中所有以点开头的文件和目录（即，所有隐藏文件和目录）。

默认情况下，Linux 是不允许文件执行的（从安全上来说，这是一件非常好的事情）。下一步我们需要做的就是允许 Shell 执行你的脚本。我们使用 chmod 命令来完成此操作，如下所示：

```
-bash-3.2$ chmod 755 my_script
```

“755”将给你读写和执行的权限，其他人将只有读和执行的权限。如果你希望你的脚本是私有的（即，只有你可以读写和执行），则请使用“700”替代。现在你就可以运行你的脚本了，只需在命令行提示符中调用脚本的文件名，如下所示：

```
-bash-3.2$ ./my_script
```

你将看到脚本的运行结果，如果脚本没有成功运行，请检查实际保存脚本的路径，然后切换到正确的目录，并尝试重新运行脚本。

1.8　小　　结

我们来总结一下，这一章我们都学习了哪些知识。

- Linux 是自由开源的类 Unix 操作系统。该操作系统的内核是由莱纳斯·托瓦兹在 1991 年 10 月 5 日首次发布。
- Linux 既可以作为服务器操作系统使用，也可以作为个人计算机的独立操作系统使用。
- Linux 内核，即 Linux 操作系统的核心。它的主要模块分以下几个部分：存储管理、CPU 和进程管理、文件系统、设备管理和驱动、网络通信，以及系统的初始化（引导）和系统调用等。
- Linux 的主要理念：一个程序只做一件事并做好、一切皆文件、小即是美、在文本文件中存储配置和数据、可移植性高于效率、简单美观。
- Linux Shell 是用户和 Linux 内核之间的接口程序，为用户提供使用操作系统的接口。
- Linux 中最常用的 Shell 有 Bourne Shell（sh）、C Shell（csh）和 Korn Shell（ksh）。
- 如果你的系统启动后进入的是文本模式，那么当你登录系统后就可以直接使用 Shell。
- Shell 脚本是使用纯文本文件，集合了一些 Shell 的语法和指令，并用正则表示法或管道命令以及数据流重导向等功能，达到我们想要的处理目的的程序。
- Shell 脚本具有使用简单、节省时间、使系统管理自动化等特点。

第 2 章　初识 Linux Shell

你是否曾考虑过，你是如何能在 Shell 中的任何目录运行一个标准的命令行功能（如 cp），而不管这个命令是否存在于哪个目录？Shell 是怎么知道这个命令存放在哪，并且是否存在的呢？

这些问题可能使刚接触 Linux Shell 环境的人感到困惑。所以在这一章中我们将认识和了解 Linux Shell 及 Shell 环境相关的一些概念。

注意：我们已经知道 Linux Shell 的类型有多种，但本书中介绍的所有 Linux Shell 相关的概念和实例都是以最常用的 Bash Shell 为基础进行讲解的。

2.1　Bash Shell

2.1.1　Bash 简介

Bash 是一个与 Bourne Shell 兼容的、执行从标准输入设备或文件读取的命令的命令语言解释器。Bash 是 Bourne-Again Shell 的缩写。

Bash 与原来的 Unix sh Shell 向后兼容，并且融合了一些有用的 Korn Shell 和 C Shell 的特性。它相对于 sh 在编程和交互式使用两方面都做了功能改进。另外，大部分的 sh 脚本可以在不做修改的情况下由 Bash 直接运行。

Bash 具有很好的移植性。它使用在构建时发现编译平台特征的配置系统，因此可以构建在几乎任一种 Unix 版本上。

2.1.2　Bash 提供的改进

Bash 的语法是 Bourne Shell 语法的一个改进版本。在大多数情况下，Bourne Shell 脚本可以被 Bash 正常地运行。下面列出了 Bash 提供的一部分改进：

- 命令行编辑。
- 命令行补全。
- 不限制命令行历史大小。
- 不限制大小的数组。
- Bash 启动文件——你可以运行 Bash 作为一个交互式登录 Shell，或作为一个交互式非登录 Shell。
- 条件表达式。

- 目录堆栈——访问目录的历史记录。
- 限制性 Shell——更多的 Shell 执行的控制模式。
- Bash POSIX 模式——使 Bash 行为更接近 POSIX 标准的规定。

2.2　Shell 在 Linux 环境中的角色

Linux 环境由以下几部分构成：

- 内核——Linux 操作系统的核心。
- Shell——为用户和内核提供一个交互的接口。
- 终端模拟器——它允许用户输入命令并在屏幕上回显命令的运行结果。
- Linux 桌面和窗口管理器——Linux 桌面是各种软件应用程序的集合。它包括文件管理器、窗口管理器、终端模拟器等等。

由此可见，Shell 在 Linux 环境中扮演了非常重要的角色，包括读取命令行、解释它的含义并执行、通过输出返回执行结果等等。

2.2.1　与登录 Shell 相关的文件

当 Linux 系统的运行级别为 3 时，用户可以从本地登录到系统控制台，或在系统运行级别为 5 时，直接以图形界面方式登录。在这两种情况下登录时你都需要输入用户名和密码。用户登录时 Bash 将会使用以下初始化文件和启动脚本：

- /etc/profile——系统级的初始化文件，定义了一些环境变量，由登录 Shell 调用执行。
- /etc/bash.bashrc 或/etc/bashrc——其文件名根据不同的 Linux 发行版而异，每个交互式 Shell 的系统级的启动脚本，定义了一些函数和别名。
- /etc/bash.logout——系统级的登录 Shell 清理脚本，当登录 Shell 退出时执行。部分 Linux 发行版默认是没有此文件。
- $HOME/.bash_profile、$HOME/.bash_login、$HOME/.profile——用户个人初始化脚本，由登录 Shell 调用执行。这三个脚本只有一个会被执行，按照此顺序查找，第一个存在的将被执行。
- $HOME/.bashrc——用户个人的每个交互式 Shell 的启动脚本。
- $HOME/.bash_logout——用户个人的登录 Shell 清理脚本，当登录 Shell 退出时执行。
- $HOME/.inputrc——用户个人的由 readline 使用的启动脚本，定义了处理某些情况下的键盘映射。

2.2.2　Bash 启动脚本

通过上一小节的介绍我们了解到，在用户登录时自动执行的脚本主要用于设置一些环境，例如设置 JAVA_HOME 的路径。其中的一些脚本被登录 Shell 调用，登录 Shell 是你

登录系统时最先执行的 Shell。它设置一些环境，然后把这些环境授予非登录 Shell。

当用户登录时，登录 Shell 会调用如下脚本：

- /etc/profile——当用户在运行级别 3 登录系统时首先运行。
- /etc/profile.d——当/etc/profile 运行时，会调用该目录下的一些脚本。
- $HOME/.bash_profile、$HOME/.bash_login 和$HOME/.profile——在/etc/profile 运行后，第一个存在的被运行。
- $HOME/.bashrc——上述脚本的中一个运行后即调用此脚本。
- /etc/bashrc 或/etc/bash.bashrc——由$HOME/.bashrc 调用运行。

当一个交互式的非登录 Shell 启动时，Bash 将读取并运行如下脚本：

- $HOME/.bashrc——如果此文件存在即被运行。
- /etc/bashrc——将被$HOME/.bashrc 调用运行。
- /etc/profile.d——此目录下的脚本将被/etc/bashrc 或/etc/bash.bashrc 调用运行。

Bash 启动脚本主要设置的环境有：

- 设置环境变量 PATH 和 PS1（我们将在 2.3.1 小节中介绍这两个变量）；
- 通过变量 EDITOR 设置默认的文本编辑器；
- 设置默认的 umask（文件或目录的权限属性）；
- 覆盖或移除不想要的变量或别名；
- 设置别名；
- 加载函数。

2.2.3　实例：定制自己的 Bash 登录脚本

本小节我们将以一个实际的.bash_profile 脚本为例，来学习如何定制一个自己的 Bash 登录脚本。首先我们在自己 Linux 账号的 home 目录下创建一个名称为.bash_profile 的文件，然后使用文本编辑器打开并编辑此文件。我们以 vi 文本编辑器为例，其内容如下：

```
$ vi ~/.bash_profile
# .bash_profile
# Custom Command Prompt
export
PS1="\n\e[1;32m[\e[0;31m\u\e[0;34m@\e[0;31m\h\e[1;32m]\e[1;32m[\e[0;34m
\w\e[1;32m]$ "

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
PATH=$PATH:$HOME/bin:/usr/bin:/usr/local/bin:/usr/sbin:/usr/local/sbin:
/sbin
export PATH
unset USERNAME
umask 022

# Custom DJRavine Modification
login pwd='pwd';
login_date='date';
```

```
login_users='users';
login_uptime='uptime';
server ip='/sbin/ifconfig | grep 'inet addr:'| grep -v '127.0.0.1' | head
-1 | cut -d: -f2 | awk '{ print $1}'';
disk available=$(df -h --block-size=1024 | awk '{sum += $4;} END {print
sum;}');
disk_used=$(df -lh --block-size=1024 | awk '{sum += $3;} END {print sum;}');
disk_size=$(df -lh --block-size=1024 | awk '{sum += $2;} END {print sum;}');
disk_available_gb=$(echo "scale=2; $disk_available/(1024^2)" | bc)
disk used gb=$(echo "scale=2; $disk used/(1024^2)" | bc)
disk size gb=$(echo "scale=2; $disk size/(1024^2)" | bc)
red="\033[31m"
blue="\033[34m"
green="\033[32m"
echo -e " "
echo -e "${blue}+-----------------------------------------------------"
echo -e " ${green}                                Welcome!                                "
echo -e "${blue}+-----------------------------------------------------"
echo -e " ${green}Server IP: ${red}"$server_ip
echo -e " ${green}Date: ${red}"$login date
echo -e " ${green}Users: ${red}"$login users
echo -e " ${green}Directory: ${red}"$login_pwd
echo -e " ${green}Uptime: ${red}"$login uptime
echo -e "${blue}+-----------------------------------------------------"
df -lh  | column -c 6 | awk '{ printf " \033[22;32m%s\t%s\t\033[22;31m%s\
t%s\t%s\n", $1, $6, $2, $3, $4,$5 }'
echo -e " ${green}Total Disk Space: ${red}${disk_size_gb} GB"
echo -e " ${green}Total Free Space: ${red}${disk available gb} GB"
echo -e " ${green}Total Used Space: ${red}${disk_used_gb} GB"
echo -e "${blue}+-----------------------------------------------------"
```

编辑完成后保存并退出文本编辑器。

再创建一个名称为.bashrc 的文件，使用文本编辑器打开并编辑此文件，内容如下：

```
$ vi ~/.bashrc
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias vi='vim'
```

编辑完成后保存并退出文本编辑器，重新登录系统，将会看到类似如下的输出结果：

```
+-----------------------------------------------------
                              Welcome!
+-----------------------------------------------------
 Server IP: X.X.X.X
 Date: Wed Jul 10 17:10:56 CST 2013
 Users: root yantaol
 Directory: /home/yantaol
 Uptime:  17:10:56 up 2 days, 17:10, 3 users, load average: 0.03, 0.03, 0.00
+-----------------------------------------------------
 Filesystem     Mounted Size    Used    Avail
 /dev/hde1      /        15G    8.0G    5.8G
 /dev/hde3      /local   208G   151G    46G
 tmpfs  /dev/shm          1.9G   0       1.9G
 Total Disk Space: 223.54 GB
 Total Free Space: 53.21 GB
 Total Used Space: 158.90 GB
+-----------------------------------------------------
```

```
 yantaol@hostname][~]$ which l.
alias l.='ls -d .* --color=tty'
      /bin/ls
yantaol@hostname][~]$ which vi
alias vi='vim'
      /usr/bin/vim
```

2.2.4　Bash 退出脚本

当登录 Shell 退出时，如果$HOEM/.bash_logout 脚本存在的话，Bash 会读取并执行此脚本的内容。此脚本主要有如下用途：

- 使用 clear 命令清理你的终端屏幕输出；
- 移除一些临时文件；
- 自动运行一些命令或脚本等。

2.2.5　实例：定制自己的 Bash 退出脚本

编辑文件~/.bash_logout，其内容如下：

```
# .bash_logout

#clear the termimal screen
clear

# clear mysql command history
echo "Clear mysql command history"
/bin/rm $HOME/.mysql_history

echo "Backup files to NAS server"
# backup files to NAS server
~/bin/backup.sh
```

编辑完成后保存并退出文本编辑器。此时我们在命令行提示符下运行 exit 命令，将会得到如下输出结果：

```
yantaol@hostname][~]$ exit
logout

Clear mysql command history
Backup files to NAS server
```

如果文件~/.mysql_history 和~/bin/backup.sh 在你的 home 目录下不存在，当运行 exit 时，会看到类似如下的输出结果：

```
$ exit
logout

Clear mysql command history
/bin/rm: cannot remove '/home/yantaol/.mysql history': No such file or
directory
Backup files to NAS server
-bash: /home/yantaol/bin/backup.sh: No such file or directory
```

2.2.6　有效的登录 Shell 的路径

/etc/shells 是一个包含有效的登录 Shell 全路径名的文本文件。这个文件会被 chsh 命令（变更你的登录 Shell）所使用也可被其他程序查询使用，比如 ftp 服务。查看/etc/shells 的内容：

```
$ cat /etc/shells
```

会看到类似如下的输出结果：

```
/bin/sh
/bin/bash
/sbin/nologin
/bin/tcsh
/bin/csh
/bin/ksh
```

你也可以使用 which 命令显示 shell 的全路径：

```
$ which bash
/bin/bash
```

2.3　Shell 中的变量

变量是任何程序或脚本的重要组成部分。变量为程序或脚本访问内存中的可被修改的一块数据提供了简单的方式。

Linux Shell 中的变量可以被指定为任意的数据类型，比如文本字符串或是数值。你也可以通过修改 Shell 中的变量来改变 Shell 的样式。

接下来就让我们来了解和学习一下 Shell 中的变量。

2.3.1　Shell 中变量的类型

Shell 中有两种变量的类型：系统变量（环境变量）和用户自定义的变量（本地变量或 Shell 变量）。

系统变量是由 Linux Bash Shell 创建和维护的变量。你可以通过修改系统变量，如 PS1、PATH、LANG、HISTSIZE 和 DISPLAY 等，配置 Shell 的样式。

常用的系统变量（环境变量）如表 2.1 所示。

表 2.1　常用系统变量表

系 统 变 量	含　义
BASH_VERSION	保存 Bash 实例的版本
DISPLAY	设置 X display 名字
EDITOR	设置默认的文本编辑器
HISTFILE	保存命令历史的文件名
HISTFILESIZE	命令历史文件所能包含的最大行数

续表

系 统 变 量	含　义
HISTSIZE	记录在命令历史中的命令数
HOME	当前用户的主目录
HOSTNAME	你的计算机的主机名
IFS	定义 Shell 的内部字段分隔符，一般是空格符、制表符和换行符
PATH	搜索命令的路径。它是以冒号分隔的目录列表。Linux 下的标准命令之所以能在 Shell 命令行下的任何路径直接使用，就是因为这些标准命令所在目录的路径定义在了 PATH 变量中，Shell 会在 PATH 环境变量指定的全部路径中搜索任何匹配的可执行文件
PS1	你的提示符设定
PWD	当前工作目录。由 cd 命令设置
SHELL	设置登录 Shell 的路径
TERM	设置你的登录终端的类型
TMOUT	用于设置 Shell 内建命令 read 的默认超时时间，单位为秒。在交互式的 Shell 中，此变量的值作为发出命令后等待用户输入的秒数，如果没有用户输入将会自动退出

当然，你可以添加上述变量到你账号的 home 目录下的初始化文件中，比如 ~/.bash_profile 文件。这样在你每次登录系统时，这些变量会被自动设置为你需要的值。

如果要查看当前 Shell 的所有系统变量，可以在控制台或终端输入如下命令：

```
$ env
```

或者

```
$ printenv
```

你将会看到类似如下的输出结果：

```
HOSTNAME=hostname
SHELL=/bin/bash
TERM=vt100
HISTSIZE=1000
USER=yantaol
LS_COLORS=no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=01;32:*.cmd=01;32:*.exe=01;32:*.com=01;32:*.btm=01;32:*.bat=01;32:*.sh=01;32:*.csh=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.gz=01;31:*.bz2=01;31:*.bz=01;31:*.tz=01;31:*.rpm=01;31:*.cpio=01;31:*.jpg=01;35:*.gif=01;35:*.bmp=01;35:*.xbm=01;35:*.xpm=01;35:*.png=01;35:*.tif=01;35:
MAIL=/var/spool/mail/yantaol
PATH=/usr/local/bin:/bin:/usr/bin
INPUTRC=/etc/inputrc
PWD=/home/yantaol
LANG=en_US.iso88591
KDE_IS_PRELINKED=1
KDEDIRS=/usr
HOME=/home/yantaol
LOGNAME=yantaol
CVS_RSH=ssh
_=/usr/bin/printenv
```

用户自定义的变量，即由用户创建和维护的变量。这一类型的变量可以使用任何有效

本书试读到此结束啦！

[Linux Shell命令行及脚本编程实例详解]

作者：刘艳涛, 编著

出版社：清华大学出版社

通过以下方式阅读更多 Powered by 书问

- 扫码分享到朋友圈可阅读更多

立即扫码

- 还不过瘾？购买书库畅读卡全本畅读此书！

查看全部书库

- 购买纸书也可畅读全本哦！

中国图书网
云书网
京东商城