

信息科学与技术丛书

石庆冬 编著

实用Linux Shell编程

- 强调动手动脑，实践与理论并重
- 不同难度的实例，适合各层次的读者
- 原创内容与程序员的开发工作紧密相关
- IT界一线人士的真知灼见



机械工业出版社
CHINA MACHINE PRESS



信息科学与技术丛书

实用 Linux Shell 编程

石庆冬 编著

机械工业出版社

本书系统地介绍了在 Linux 系统中广泛使用的 Bash 脚本语言的基本知识。全书内容的安排由浅入深、体系合理。先讲解脚本的概念和学习环境的搭建,接下来介绍 Linux 的常用命令,然后根据概念之间的依赖关系,讲解 Bash 环境设置、变量与数组、条件流程控制、循环、函数、正则表达式、文本处理、进程与作业、高级话题等。

本书适合 Linux 脚本编程零基础的读者学习。可以作为高校、职校计算机专业的参考教材,也可以作为从事 Linux 软件相关工作的工程师的参考读物或培训教材,同时还是广大 Linux 脚本爱好者不可多得的一本中文原创读物。

书中代码可在 <http://www.cmpbook.com> 免费下载。

图书在版编目 (CIP) 数据

实用 Linux Shell 编程/石庆冬编著. —北京:机械工业出版社, 2014. 11
(信息科学与技术丛书)

ISBN 978-7-111-48202-4

I. ①实… II. ①石… 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 230876 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:车 忱

责任校对:张艳霞

责任印制:李 洋

北京振兴源印务有限公司印刷

2014 年 11 月第 1 版·第 1 次印刷

184mm×260mm · 19.25 印张·476 千字

0001—3000 册

标准书号:ISBN 978-7-111-48202-4

定价:52.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版



出版说明

随着信息科学与技术的迅速发展，人类每时每刻都会面对层出不穷的新技术和新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术不断地受到挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机和互联网在社会生活中日益普及，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学与技术 in 电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络和工程应用等内容，注重理论与实践的结合，内容实用、层次分明、语言流畅，是信息科学与技术领域专业人员不可或缺的参考书。

目前，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设做出贡献。

机械工业出版社

前言

计算机操作系统主要分为 Windows 和 UNIX/Linux 两大阵营，IT 技术人员对这两种操作系统都应该熟悉。学习免费的、开源的、能在 PC 上运行的 Linux 操作系统的必要性无需赘言。学习 Linux 都是从执行键盘字符命令的地方——shell（外壳）开始。shell 包裹着 Linux 的内核和硬件，并提供了一套用于和内核通信的指令集，shell 本身也是一门脚本程序设计语言。Linux 下包含多种 shell，而 Bash 是 Linux 的默认 shell，所以掌握 Bash 编程是深入学习 Linux 的前提条件。

本书深入浅出、系统地介绍了在 Linux 系统中进行 Bash 编程的基本知识。

第 1 章介绍脚本的概念、学习环境的搭建、Linux 命令的格式和获得命令帮助的途径。

第 2 章介绍常用的 Linux 命令和基础知识，包括软链接和硬链接、权限控制、SUID 与 SGID、文件的三种时间和 vi 的使用等。

第 3 章介绍 Bash 环境的设置和几个常用的内置命令，还介绍了管道、输入输出重定向和文件描述符等概念。本章还讲解了如何写出一个可执行的脚本。

第 4 章介绍变量与数组。包括变量的定义、变量的读取、变量的导出、只读变量、整型变量的运算、索引数组、关联数组、环境变量、内置变量和字符串的操作等。

第 5 章讲解条件流程控制。包括整型数关系运算、字符串关系运算、文件属性条件判断、逻辑的与或非、命令的与或非、if 与 case——两种控制程序流程的命令结构，最后介绍了 null 命令的应用。

第 6 章讲解循环。包括 for、while 和 until 循环，控制循环的命令 break、continue 和 shift。还讲解了产生字符菜单选项的 select 命令和用来处理脚本选项和参数的 getopt 命令。

第 7 章讲解函数。包括函数的定义、函数参数的处理、函数内的局部变量、函数的导出、函数的返回和递归函数等。

第 8 章介绍正则表达式的概念和文本处理的常用方法。包括过滤命令 grep 和 egrep、剪切内容命令 cut、合并行内容命令 paste、转换或删除字符命令 tr、排序命令 sort、流编辑器 sed、文本扫描和处理语言 awk 和格式化打印命令 printf。

第 9 章讲解进程与作业。包括进程与作业的查看、挂起作业、将作业放置到前台或后台、给作业发送信号的命令 kill、捕获信号的命令 trap、等待命令 wait、移除作业命令 disown 和暂停 shell 的命令 suspend 等。

第 10 章介绍一些 Bash 中的高级应用。包括二次扫描命令 eval、目录栈命令、并行命令 parallel、（非）登录与（非）交互 shell 的概念和 shell 的选项（包括 Bash 自带的选项和可通过 set 与 shopt 命令设置的选项）。最后，介绍了脚本调试的基本方法和几个脚本实例。

附录简单介绍了 Bash 的 shellshock 漏洞。

本书强调实践。每讲解一个命令或概念时，几乎都是通过例子引出或者用实例解释，而不是将其官网手册上的选项和参数列表的中文翻译全盘给出。本书基于 Ubuntu 操作系统，所有的例子都在 Ubuntu 下经过了验证。

本书作者在大型企业从事软件工作多年。在学习和使用 Bash 的过程中，作者发现：有关 Bash 的资料、译著较多，网络上志愿者们的心得体会或转载的资料很多，但中文的、原创的图书比较少。本书对于作者而言是多年学习和使用 Bash 的一个总结，现在整理成册并出版，一方面可以帮助他人学习，为大家奉献一本 Bash 中文原创读物；另一方面是对自己学习效果之检查和进一步学习 Bash 的鞭策。

原中国煤炭报总编室主任谢国实教授、诺基亚资深工程师邹德禹、邱海泓，对本书进行了非常认真的审阅，并提出了很多中肯的修改意见，在此深表感谢！

希望广大读者朋友在阅读过程中，如果发现缺陷和不妥之处，能提出宝贵的建议和意见。您的反馈，很可能就是此书将来再版的依据和动力！

作者

2014 年春于北京

目 录

出版说明

前言

第 1 章 概述	1
1.1 为什么要学习 Linux shell 脚本	1
1.2 什么是 shell	1
1.3 什么是 shell 脚本	2
1.4 为什么要学习 Bash	3
1.5 Bash 学习环境的准备	4
1.5.1 Linux 的准备	4
1.5.2 Cygwin 工具	4
1.5.3 VMware 与 Linux 虚拟机	7
1.6 Linux 命令格式简介	10
1.7 如何获得 Linux 命令的帮助	11
第 2 章 Linux 基础知识与常用命令	15
2.1 Linux 的文件与目录结构	15
2.2 查看文件清单命令 ls	16
2.3 浏览文件命令 cat、more、less、head 和 tail	18
2.4 文件统计命令 wc	19
2.5 改变当前工作目录命令 cd	20
2.6 创建目录命令 mkdir	23
2.7 复制命令 cp	24
2.8 重命名或移动命令 mv	25
2.9 创建符号链接和硬链接命令 ln	26
2.10 显示当前目录命令 pwd	30
2.11 产生空文件或者改变文件时间戳命令 touch	30
2.12 查看账户名及其所属组的命令 whoami、id 和 groups	32
2.13 文件与目录的权限	32
2.14 改变权限命令 chmod	35
2.15 SUID 与 SGID 以及粘滞位	37
2.15.1 SUID	37
2.15.2 SGID	39
2.15.3 粘滞位	41
2.16 查看文件的三种时间	43
2.17 删除命令 rm 与 rmdir	44
2.18 编辑文件命令 vi	45
2.19 用于显示的命令 echo	48

2.20	查看文件类型命令 <code>file</code>	49
2.21	显示树状目录信息命令 <code>tree</code>	50
2.22	查找命令 <code>find</code>	51
第 3 章 Bash 内置命令与环境简介		53
3.1	内置命令与外部命令	53
3.2	认识 Bash 环境	58
3.2.1	命令行提示 <code>PS1</code>	59
3.2.2	搜索路径 <code>PATH</code>	60
3.2.3	设置和取消别名命令 <code>alias</code> 和 <code>unalias</code>	61
3.2.4	修改 <code>.bashrc</code> ——设置自己的环境	62
3.3	权限掩码命令 <code>umask</code>	64
3.4	<code>source</code> 命令和点命令	66
3.5	编写并运行第一个脚本	67
3.6	命令解释顺序与命令类型的查询	68
3.6.1	命令的解释顺序	68
3.6.2	改变命令解释顺序的三条内置命令	69
3.6.3	命令类型的查询	70
3.7	命令的退出状态以及命令 <code>true</code> 和 <code>false</code>	71
3.8	管道	72
3.9	输入输出重定向	73
3.9.1	标准输入和标准输出	73
3.9.2	输入输出重定向与输出追加重定向	74
3.9.3	预防输出重定向覆盖旧文件	77
3.9.4	标准错误输出与“黑洞”	78
3.9.5	同时把结果输出至标准输出与文件的命令 <code>tee</code>	80
3.10	<code>exec</code> 命令与文件描述符	82
3.11	有关命令的进一步讨论	87
3.11.1	一行多命令	87
3.11.2	将命令放在后台执行	87
3.11.3	命令行的续行	88
3.11.4	命令的补齐与命令历史	89
3.11.5	操作名字含空格的文件	90
3.11.6	操作名字首字母为减号的文件	90
3.11.7	Windows 与 Linux 文件格式的互转	91
3.11.8	小括号与大括号中的命令	92
3.11.9	子 shell	92
第 4 章 变量与数组		94
4.1	变量的定义与清除	94
4.2	字符串定义及单双引号与大括号的使用	95

4.3	将命令执行结果赋给变量——反引号与 $\$()$	98
4.4	键盘输入变量值——内置命令 read	100
4.5	整型变量运算	102
4.6	浮点运算	106
4.7	定义只读变量命令 readonly 和 declare -r	107
4.8	定义整型变量命令 declare -i	107
4.9	数组	109
4.10	关联数组	113
4.11	导出变量命令 export 和 declare -x	114
4.12	declare 命令总结	116
4.13	环境变量与特殊变量	116
4.14	内置变量	120
4.15	计算表达式值的命令 expr	125
4.16	变量测试及其相应的赋值	127
4.17	字符串操作	132
4.17.1	Bash 内置的字符串操作	132
4.17.2	用命令 expr 处理字符串	135
第 5 章 条件流程控制		138
5.1	条件判断与 test 命令	138
5.1.1	整型数关系运算	138
5.1.2	字符串关系运算	139
5.1.3	文件属性条件判断	141
5.1.4	逻辑的与或非	143
5.1.5	与或非的优先级	145
5.1.6	双中括号格式	145
5.1.7	在双小括号里面进行整数比较	147
5.1.8	命令的与或非	148
5.1.9	判断变量是否定义	150
5.2	条件测试结构 if	151
5.3	if-else 结构	153
5.4	if-elif 结构	156
5.5	分情况选择处理——case 命令	159
5.6	命令 exit 与 if 及 case 命令的配合	162
5.7	用 here 文档与 case 命令生成菜单	163
5.8	null 命令	165
第 6 章 循环		169
6.1	for 循环	169
6.2	算术 for 循环	175
6.3	while 循环	177

6.4	until 循环	179
6.5	用 break 和 continue 控制循环	181
6.6	用命令 shift 控制循环	185
6.7	选择命令 select	187
6.8	循环命令与 I/O 重定向及管道的配合	192
6.9	脚本的选项与参数	195
6.9.1	命令 shift	195
6.9.2	命令 getopts	197
第 7 章	函数	202
7.1	函数定义	202
7.2	给函数传递参数	203
7.3	函数的局部与全局变量	204
7.4	当前的函数名 FUNCNAME	206
7.5	在命令行执行函数	207
7.6	查看当前 shell 的函数定义	208
7.7	选项-f 与函数的导出、清除与只读设置	209
7.8	返回命令 return	210
7.9	递归函数	213
第 8 章	正则表达式与文本处理	215
8.1	文件名替换	215
8.1.1	多字符替换*	215
8.1.2	单字符替换?	216
8.1.3	范围替换[]与[!]	217
8.2	正则表达式与 grep	218
8.2.1	过滤器 grep	218
8.2.2	扩展的 egrep	224
8.2.3	POSIX 字符类	224
8.2.4	Bash 扩展模式匹配	226
8.3	剪取内容命令 cut	227
8.4	合并相应行的命令 paste	229
8.5	转换或删除字符命令 tr	231
8.6	排序命令 sort	233
8.7	流编辑器 sed	236
8.7.1	替换命令 s	236
8.7.2	其他命令	240
8.7.3	一行多条命令与保存匹配&	242
8.7.4	sed 的退出状态	243
8.7.5	sed 脚本	244
8.8	文本处理工具 awk	245

8.9	格式化打印命令 printf	251
第 9 章	进程与作业	254
9.1	查看进程命令 ps	254
9.2	挂起进程 (Ctrl+Z) 键	255
9.3	前台 fg 与后台 bg	256
9.4	发送信号命令 kill	257
9.5	等待命令 wait	259
9.6	捕获信号命令 trap	261
9.7	移除作业的命令 disown	264
9.8	暂停 shell 的命令 suspend	265
第 10 章	高级话题	267
10.1	二次扫描命令 eval	267
10.2	目录栈操作命令 pushd、popd 与 dirs	269
10.3	波浪号扩展	272
10.4	(非) 登录及 (非) 交互 shell	273
10.5	Bash shell 选项	273
10.6	用命令 set 设置 shell	274
10.7	用命令 shopt 设置 shell	278
10.8	终端行设置命令 stty	280
10.9	不在脚本和函数内使用别名	282
10.10	Bash 调试	284
10.11	并行命令 parallel	287
10.12	模拟旋转型进度指示	289
10.13	删除文件空行	291
10.14	完善 while-shift 循环	292
附录	Bash 安全漏洞 shellshock	294
	参考文献	297

第1章 概 述

Linux 在近些年发展和普及非常迅速，它是一个可以安装在个人电脑上的类 UNIX 操作系统。Linux 的魅力在于它的免费、开源、与互联网的结合，还有在 Linux 下众多的应用程序。Linux 的用户越来越多，甚至挤占了 UNIX 的生存空间，这已经是不争的事实。可以说 Linux 无处不在。Linux 基金会执行委员 Jim Zemlin 说过，“我们每天都在用着 Linux，即使大家不知道。Linux 显然已成为我们生活中不可或缺的一部分了。”

可能有人说他只是用安装了 Windows 系统的个人电脑上网，并未接触到 Linux。他确实没有直接接触 Linux，但他在利用互联网查资料或者与他人沟通时，很可能有他看不到的数台 Linux 服务器在为他工作着。举个大家看得到的例子吧，近几年 Android（安卓）平台手机很流行。Android 就是一种以 Linux 为基础的开源操作系统，主要用于便携设备。假设让全世界使用 Linux 或者与 Linux 相关的设备停止工作一分钟，那么将会有手机无法接打、交通信号灯不工作、股市交易停止、网站瘫痪、核潜艇不听指挥、工厂停工等现象发生。

1.1 为什么要学习 Linux shell 脚本

Linux 是许多工程技术和科研人员必须掌握的基础知识。当前的 Linux 发展得像 Windows 一样，似乎用鼠标和图形界面就可以完成许多工作。

有人说，“我使用过某公司开发的 Linux 下的某图形界面工具，用起来简单，功能强大，单击几个按钮，该工具就可做许多事情。”的确，按一按按钮即可。但是，每单击一下按钮，都可能触发了某个脚本的运行，或者是一个以上的，甚至是一种以上的脚本的执行。因为图形界面工具的开发完成了许多脚本编码工作，才使得用户得到了简单易用的软件产品。

对于想比较深入地了解 Linux 的人而言，如果只会执行几个常用的 Linux 命令，或者只能在图形 Linux 系统中单击鼠标是不够的。试想，用鼠标加上图形界面如何实现条件判断功能？如何实现嵌套的循环？学习 shell 脚本对于深入地学习 Linux 是很有帮助的，是必须的。一句话，shell 脚本是想深入学习或全面了解 Linux 的人的必修课。

1.2 什么是 shell

shell 是 Linux 的外壳，是 Linux 的用户界面，可以为用户提供输入命令和参数，并可以得到命令执行结果的环境。用户、shell 和操作系统之间的关系如图 1-1 所示。

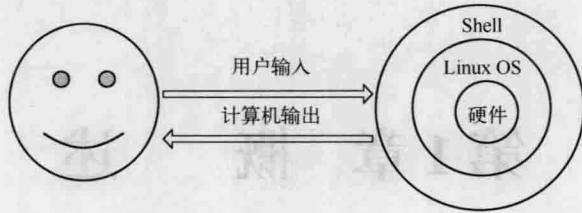


图 1-1 Shell 是包围在 Linux 系统之外的一层外壳

例如，用户想知道现在的时间，打开一个 shell 命令行窗口，也叫 shell terminal（本章后面将讲述 shell 环境的准备），输入命令 `date` 即可。

```
$ date
Sun Jul 29 11:06:30 EDT 2012
```

看来 `date` 命令提供的信息不少，时区也显示出来了，EDT 表示美国东部夏令时间 (Eastern Daylight Time)。如果只想知道日期或者时间，不希望有其他信息输出，需要输入 `date +%F` 或者 `date +%T`。

```
$ date +%F
2012-07-29
$ date +%T
11:12:54
```

再举两个例子。用户想查看当月的日历 (`calendar`)，输入命令 `cal` 即可。

```
$ cal
          July 2012
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

如果用户想计算 `4+5` 并打印其结果，可运行 `echo $((4+5))` 命令。

```
$ echo $((4+5))
9
```

根据以上几个例子可以看出，shell 就是输入 Linux 指令并得到计算机输出的地方。Linux kernel 是统一的，但是 shell 有很多种，比较常用的有 Bash (Bourne Again shell)、Csh(C shell)、Tcsh(TENEX C shell)、Ksh(Korn shell)、Zsh(Z shell)等。掌握了一种 Linux shell 之后，学习其他 shell 非常容易。

1.3 什么是 shell 脚本

可执行程序一般有两种实现方式：一种是二进制方式；另一种是脚本 (script) 方式。

二进制方式是先将编写好的程序（如 C/C++ 程序）进行编译，变成计算机可识别的指令代码（如 .exe 文件），然后再执行。这种编译好的程序只能执行、使用，却看不到它的程序内容。

脚本，简单地说就是由一条一条的命令组成的文本文件，可以用记事本查看或者编辑。脚本程序在执行时，是由系统的一个解释器将其一条条地翻译成计算机可识别的指令，并按程序的逻辑顺序执行（一般来说，脚本程序是确定的一系列控制计算机进行运算操作的动作的组合，在其中可以实现一定的逻辑分支等）。因为脚本在执行时多了一道翻译的过程，所以它比二进制程序执行效率要稍低一些。

Linux shell 脚本，类似于 DOS 批处理（batch）脚本，就是将一些可执行的 Linux 命令放在一个文件里面，然后按照顺序（确切地说，是逻辑顺序，因为会有判断、分支、循环等），命令一条条地执行。例如，接着前面的例子，假设需要同时知道现在的日期、时间和本月的日历，那么需要先后输入相应的 3 条命令，这并不麻烦。但是假设需要知道很多信息，需要输入 30 条命令或者更多，并且每天都要执行同样的命令，是不是就有些麻烦了？如果把这些命令写在一个文本文件里面就不麻烦，该文本文件就是一个脚本。每天只需执行该脚本，写在脚本里面的命令会依次运行。还是接着前面的例子，这 3 条命令组成的脚本如下：

```
#!/bin/bash
date +%F
date +%T
cal
```

该脚本一共 4 行，第一行指明了脚本解释器的位置，接下来的 3 行依次是需要执行的命令。运行该脚本就会得到想要的结果。运行一个脚本，输入脚本文件名即可，仅需要用键盘输入一条命令；如果没有脚本，将需要输入 3 条、30 条或者更多的命令。

但 shell 脚本不能被简单地理解为：将某些 Linux 命令“堆放”在一起。实际上，每种 shell 脚本都是一门程序设计语言。一个包含多条命令的脚本，打开它，从表面上看，命令被“堆放”在一起，但是命令的先后顺序不是任意的，一般是有依赖关系的，有一定逻辑的。

与 C/C++，Java，Fortran，Pascal 等程序设计语言相比，shell 脚本语言掌握起来要相对容易。Linux shell 没有复杂和丰富的数据类型，虽然 Linux shell 包含分支、判断、循环等功能，但 shell 脚本在本质上的确就是将某些 Linux 命令“堆放”在一起（shell 脚本运行时，当中的命令依次接受脚本解释器的解释，例如，一个包含多条命令的脚本，即便最后一条命令有语法错误，前面的命令仍然可以照常运行；即便中间某条命令没有正常运行，假设后面的命令不依赖于它，后面的命令仍然可以照常运行）。所以很难看到逻辑非常复杂的 shell 脚本，看懂别人的 shell 脚本通常不是一件难事。

1.4 为什么要学习 Bash

第一个主流的 shell 是 Bourne shell，简称 sh，以发明者 Stephen Bourne（史蒂夫·波恩）的姓来命名。1979 年，sh 随着 UNIX 版本 7 发布并开始流行。

Bash 诞生于 1987 年，它是 sh 的超集，几乎完全兼容 sh 并拥有更多的特性和功能。在 UNIX 系统里，程序 sh 与 Bash 是不同的；而在 Linux 系统里，也有一个名字为 sh 的程序，

但它不是真正的 sh，而是 Bash，因为 Linux 里面的 sh 是一个指向程序 Bash 的符号链接，这个符号链接的设置是为了使 sh 脚本在 Linux 下不做修改就可运行。读者如果有机会看到较旧的脚本，会发现第一行大多是#!/bin/sh。有时，遇到年纪较大的工程师，会发现他在写 Linux 脚本的第一行时，仍然习惯写#!/bin/sh。

Csh 的语法有点类似 C 语言，这也是其名字叫 Csh 的原因。Tcsh 是 Csh 的超集。因为 Tcsh 兼容 Csh，所以多数用户选择功能更强的 Tcsh，使用 Csh 的人在逐渐减少，并且越来越多的 Linux 系统的 Csh 被设置为指向 Tcsh 的符号链接。Tcsh 也很重要，而且流传很广，但是 Tcsh 有个致命的弱点：没有函数功能。

Bash 是几乎所有 Linux 操作系统的默认的 shell。有时会发现某种 Linux 系统默认没有安装 Tcsh、Ksh 或 Zsh，但是，要发现某类 Linux 系统默认没有安装 Bash 是一件非常难的事情。Bash 简单易学，功能强大，并借鉴了 Ksh 和 Csh 很多有用的特性。如果掌握了 Bash，再要掌握其他种类的 shell，将会很容易。因此学习 Bash 是明智的选择。

1.5 Bash 学习环境的准备

1.5.1 Linux 的准备

细分 Linux 操作系统，有上百种之多。常用的 Linux 有 Redhat、Fedora、SUSE、Debian、Ubuntu 等。Linux 的多数命令，在各种 Linux 系统上没有本质差别。比较常用的命令及常用选项和参数，在各种 Linux 系统上几乎完全相同。如果某脚本在一种 Linux 系统上可以运行，在其他 Linux 系统上一般也可以正常运行，或者只需小小的改动即可运行。

如果读者已经有 Linux 服务器或 PC，并且知道了自己的账户名和密码，那么就已经有 Bash 的学习环境，因为前面说过，Bash 是 Linux 系统的默认 shell。如果没有现成的 Linux 系统，可以自己试着安装和搭建。Linux 的安装盘不需要购买，上网任选一种 Linux，下载即可。以 Fedora 为例，http://fedoraproject.org/zh_CN/get-fedora#desktops 为下载链接。下载光盘镜像文件即 ISO 文件之后，可以将其制作成安装盘。制作安装盘的步骤和安装的步骤，在 get-fedora 网页上可以很容易找到。

可以将自己的计算机安装为 Linux 操作系统。或者将计算机安装为双系统，在启动的时候选择进入 Linux 或 Windows，对于只有一台计算机的用户来说，也许要频繁地重启计算机。总的来说，安装 Linux 比较容易，但对于初学者还是有一些难度。接下来的两小节提供了两种稍微简单的搭建 shell 学习环境的方法，并且可以使只有一台计算机的用户在 Windows 操作系统下“无重启”地学习 Linux shell 编程。

1.5.2 Cygwin 工具

Cygwin 是一个在 Windows 平台上运行的模拟 Linux/UNIX 环境的免费软件工具。它的下载、安装与使用，详见 <http://www.cygwin.com/>。

Cygwin 的下载和安装不难，大致过程为：先在 <http://www.cygwin.com> 下载一个名为 setup.exe 的可执行程序（目前新版本为 setup-x86.exe），运行该程序，然后单击 Next>按钮，得到 Cygwin 下载及安装的主界面，如图 1-2 和图 1-3 所示。

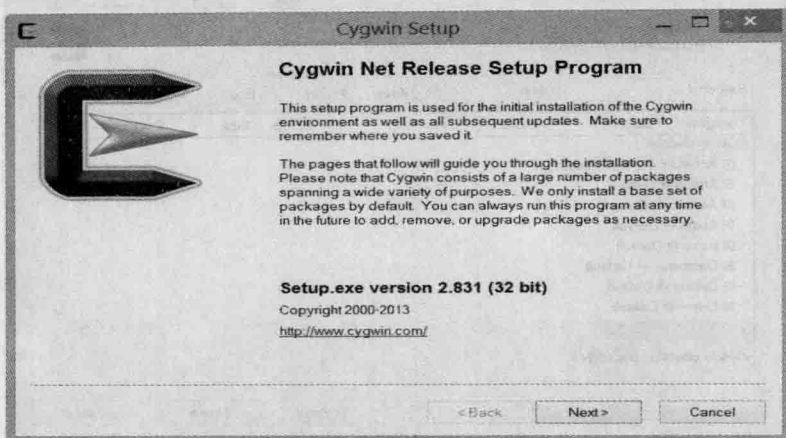


图 1-2 运行 Cygwin 的 Setup 程序

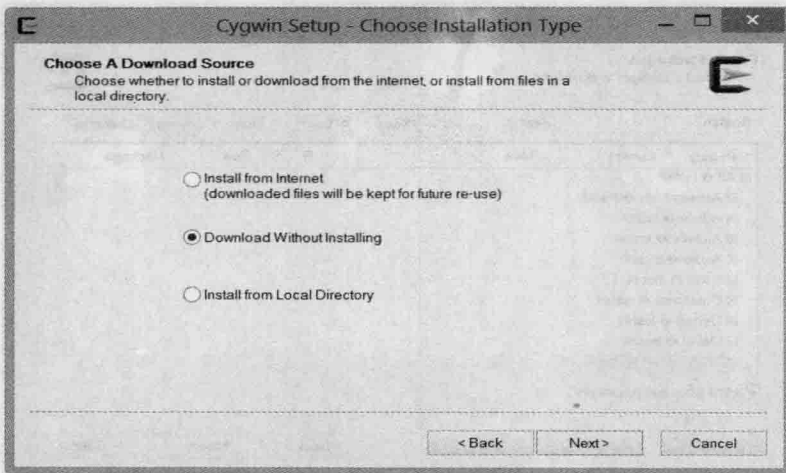


图 1-3 Cygwin 的下载及安装主界面

建议先下载到本地后再安装，就是下载的时候选择图 1-3 中的 **Download Without Installing**。Cygwin 不是一个很小的软件，下载和安装的时间也不是很短。除非网速很快，网络非常稳定，才可以选择 **Install from Internet**（从网络安装）。

下载的时候，可以选择默认下载，也可选择全部下载。如图 1-4 所示的多个“Default”就是默认下载，用鼠标单击最上面的 **All@Default**，所有的 **Default** 变为 **Install** 后，就是全部下载，如图 1-5 所示。下载完成后，再次运行 **setup.exe**，选择从本地安装，也就是选择图 1-3 中的 **Install from Local Directory**。安装包括默认安装和全部安装。

对于 Linux 下的各种应用不是很熟悉的人，如果硬盘空间足够大，在下载和安装的时候，最好选择全部下载和全部安装。全部下载和安装的用时较长，如果想快速熟悉 Cygwin 的下载、安装和基本使用，可以先选择默认下载和默认安装。

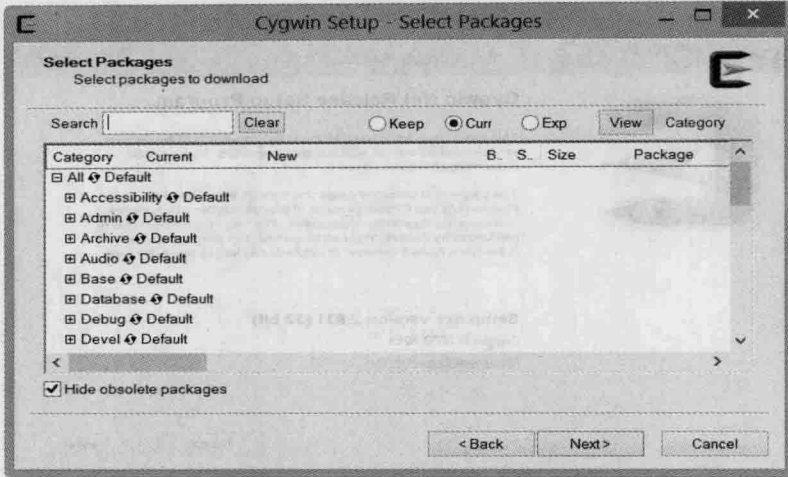


图 1-4 Cygwin 的默认下载

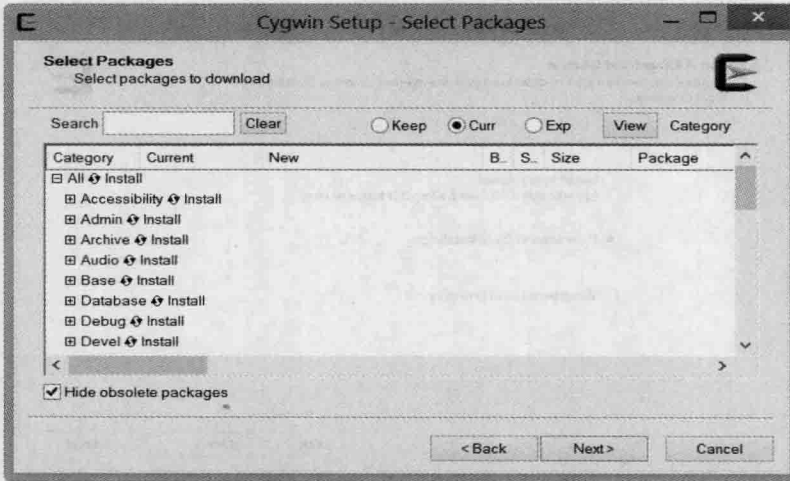


图 1-5 Cygwin 的全部下载

安装完成后，在桌面可以看到如下的图标：



双击该图标，即可进入 Cygwin 命令行字符界面（也叫 CLI, Command-Line Interface），会看到一个闪动的光标。这时所见的就是 Cygwin 默认的 shell 的“外表”，如下所示：

```
user@earth-PC ~
$
```

简单认识一下 Cygwin 界面：user 为账户名（account name），就是登录 Windows 的账号，earth-PC 为计算机名，\$ 为命令行提示符。Cygwin 的默认 shell 为 bash，运行 echo \$0 便可知。命令 echo 用来显示一行内容，或者显示变量的值；\$0 是一个内置变量，表示 shell 的名字（如，bash、csh、tcsh 等）或者 shell 脚本的名字。