

免费访问交互式UNIX培训网站



实践大师： UNIX SHELL编程篇

UNIX SHELL PROGRAMMER'S INTERACTIVE WORKBOOK

[美] CHRISTOPHER VICKERY 著

李祥凯 李国华 王正强 等译

李祥凯 审校

- ▶ 掌握UNIX shell编程!
- ▶ 包含最新的KornShell内容
- ▶ 为系统管理员新添shell编程
- ▶ 通过实践、练习及课外自修项目学习



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

实践大师:UNIX SHELL 编程篇

UNIX SHELL PROGRAMMER'S INTERACTIVE WORKBOOK

[美] CHRISTOPHER VICKERY 著

李祥凯 李国华 王正强 等译

李祥凯 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

Authorized translation from the English language edition published by Prentice – Hall, Inc.

Copyright © 1998

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry, China.

Copyright © 2000

本书中文简体专有翻译出版权由美国 Prentice – Hall, Inc. 授予电子工业出版社。该专有出版权受法律保护。

图书在版编目(CIP)数据

实践大师:UNIX SHELL 编程篇/(美)维克里(Vickery, K.)著;李祥凯等译. – 北京:电子工业出版社. 2000.1

书名原文:UNIX SHELL PROGRAMMER'S INTERACTIVE WORKBOOK

ISBN 7-5053-4432-3

I. 实… II. ①维… ②李… III. Unix 操作系统-程序设计 IV. TP316

中国版本图书馆 CIP 数据核字(1999)第 73171 号

书 名:实践大师:UNIX SHELL 编程篇

原 书 名:UNIX SHELL PROGRAMMER'S INTERACTIVE WORKBOOK

著 者:[美]CHRISTOPHER VICKERY

译 者:李祥凯 李国华 王正强 等

审 校 者:李祥凯

责任编辑:周宏敏

特约编辑:唐 雅

排版制作:电子工业出版社计算机排版室监制

印 刷 者:北京天竺颖华印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:24.25 字数:621 千字

版 次:2000 年 1 月第 1 版 2000 年 3 月第 2 次印刷

书 号:ISBN 7-5053-4432-3

TP·2064

印 数:2000 册 定价:46.00 元

版权贸易合同登记号 图字:01-1999-3362

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

译者的话

UNIX 系统从其在 AT&T 贝尔实验室踏入计算机世界以来,经过三十多年的发展已经成为当今世界主流的操作系统之一,目前虽然在市场上面临 Microsoft Windows 的强有力的竞争,但是它仍然是从笔记本电脑、PC、PC 服务器、中小型机、工作站、大巨型机及群集、SMP、MPP 上全系列通用的操作系统。特别是近几年来 Linux 的飞速发展,已逐步形成了与 Microsoft Windows 抗衡的势头,它在短短的八年的时间内在全球的装机容量超过 1000 万台,并且这一发展趋势每年以较大的比例增长。可以这样说在 PC 市场上 Microsoft Windows 将面临着 Unix 的巨大挑战,Unix 将占领 PC 市场较大的市场份额以不再是神话。

在 UNIX 应用日趋广泛的今天,对于我们每一个从业人员来说,快速并且有效地学习和掌握一门 UNIX 技术将增加您就业的砝码。因而在学习过程中有一本好的参考书是至关重要的,它往往起到事半功倍的效果。UNIX 实践大师这套丛书就是为了让读者迅速入门并能快速掌握 UNIX 各项技术而编写的。它充分利用了书本和计算机的相互性以及当前的计算机网络技术,从基本的定义入手,然后结合练习并可以在 Internet 上同不同的用户展开深入的讨论来对所学的知识进行巩固和提高,从而充分提高了学习的趣味性和效率。

《UNIX 实践大师—shell 程序员篇》这本书主要讲述与 shell 编程的有关的各项技术,书中对 shell 的设置、命令语法、变量的用法、参数、基本测试、循环、模式、扩展和置换以及输入/输出和陷阱处理等进行由浅至深的阐述,结合实践给出了一些用法的实例以及容易出现的问题分析,并在最后给出了自测题和思考题供读者检查学习情况使用。本书适合于具有一定 UNIX 使用经验的编程人员和系统维护人员使用,也可作为 UNIX 高级培训的教程。

本书是由李祥凯、李国华和王正强等同志翻译,最后由李祥凯同志对全书进行了审校。参加翻译工作的同志还有:籍顺心、李波、李勇、陈爱军、黄洁、李涛、李洁、赵燕花。由于本书信息量大,参与翻译的人员手头又有许多开发任务,为了尽快把这本书奉献在读者面前,急切之中不妥之处在所难免,敬请广大读者不吝赐教。

总 编 的 话

Prentice Hall 的《UNIX 实践大师》一书是专门为使读者迅速入门并迅速提高而编写的,同时它也是一本很好的速查手册,能随时查询所需要的信息。

我们确认读者将理解我们的这种直截了当的学习方法。本书的每章开头部分将对本章的学习目标给出一个清楚的定义。一系列的实践将构成每一章的核心。每个实践将会以练习的形式教给读者以专门的技巧。读者将上机完成这些练习,并回答提出的问题,并就答案进行更深入的讨论和探索。每个实践以一个多项选择的自检题结束,来加强所学的知识。最后在每章的结尾包含了一些思考题。这些思考题将通过本章的学习,检验读者综合运用技巧的能力。

我们的目标是使学习更具魅力,使读者成为一个高效多能的学者。

读者并非单独学习,因为每本书拥有一个“伙伴网址”,每个网址允许读者查询本书所讨论概念的更详细的信息。附加的自检题能更进一步地提炼出本章知识的精华,也许更重要的是,在那儿可能有用户查找同样的问题。

本丛书所有的“伙伴网址”在 <http://www.phptr.com/phptrinteractive> 上均能找到。

Mark L. Taub

总编辑

Prentice Hall PTR Interactive

简介

专业和业余的程序员可能正在寻找类似于 KornShell 的脚本编程语言。本书将主要讲述 KornShell 语言,因为它是有价值的工具,可以用最少数量的代码开发出功能强大的应用程序。

KornShell 既是一种脚本语言又是一种命令解释器,这就意味着不仅可以使 shell 编程的特征又可同工作中处理程序的命令混合使用。UNIX 称这种命令解释器为 shell,在 UNIX 系统中除了 KornShell 外还有很多 shell: Bourne shell(sh)、Bourne - again shell(bash)、C shell(csh)以及 TC shell(tcsh)等等,Microsoft Windows 在 Windows 95/98(command.com)和 Windows NT/2000(cmd.exe)的命令解释器中也提供了基本的脚本特征。KornShell 完全具有 POSIX 标准的优点,并且实用的平台范围更加广泛,不仅包含所有的主要的 UNIX 系统,而且在 Microsoft Windows 系统中也同样适用。作为一种 KornShell 程序员的开发技巧也可拓展到其他编程平台上。

本书使用要求

要使用这本书,计算机上需要安装最近的 KornShell 版本。第 1 章中的内容将会有助于确认系统的设置是否正确,还包括如何检查系统中是否安装了正确的版本,以及在有的时候如何找到这个版本(在没有安装的情况下如何获得拷贝)。操作系统可以是主要的 UNIX 版本,包括 Linux,也可以是 Microsoft Windows,在这种情况下就需要使 UNIX 的程序能够在 Windows 上运行的软件包,特别是 U/WIN 软件包,它是由 AT&T 实验室的 David Korn(《KornShell》的作者)开发的。如果工作在 Windows 环境下,并且可以选择其中的某个版本,应该选择 Windows NT/2000,因为它比 Windows 9x 版本提供了支持更多的 shell 的特征。

同时也需要 Internet 访问本书的“伙伴网址(web site:网站)”的主页,因为它也是本书完整的一部分。

本书的组织

不要仅仅只是阅读这本书,这是一本交互式的工作手册,应交互地阅读本书并使用计算机。每一章都有一个或多个实践,每个实践都有一个或多个练习,比较典型的是在计算机上运行了某个命令后,练习就提出了一系列需要去回答的问题。所有问题的答案均在每个实践的最后给出,并且还跟着解释用到的概念。即使能够回答这些问题,在进入下一部分之前也要检查一下这些答案,因为答案中用到的解释材料在回答以后的问题中可能需要。最好放一个书签以便在需要用时很容易找到。

在每个实践的末尾有一系列的自测题,可以用来检查是否真正掌握了这部分内容。这些问题的答案在附录 A 中给出。在本书的伙伴网址的主页里每一章还有一些附加的实践题。

第 1 章主要讲如何设置计算机,在这之后的其余章节中将介绍 KornShell 编程,目的是有

一个逻辑的次序,选择这种方法来组织也是由于教育学的原因。但是应该意识到它具有这样一个特征:如果知道是晚些时候来的材料,就应该同早一些的材料区别对待。目的是把重点放在学习语言部分,而不要为了使每个练习成为完全实际的任务而去介绍无关的材料。本书后面的章节给出了更多实际的问题以及解题的方案。

附录 B 是所有 KornShell 的内部命令的字母引用(reference:参考)指南,是从 KornShell 的权威性的参考书——《The New KornShell Command and Programming Language》中抽取出来的,这本书由 Morris I. Bolsky 和 David G. Korn(Prentice - Hall 1995)编写。本书尽可能多地包含 KornShell 的内容,但仍有部分语言不适合在本书中做出介绍。在完成本书时,如果还想学习更多的内容,应该看一下 Bolsky 和 Korn 的书。如果对用 UNIX 系统的图形用户接口开发 KornShell 程序感兴趣,可以参考 J. Stephen Pendergast, Jr. (Addison - Wesley, 1995)所编著的《Desktop KornShell Graphical Programming》一书。

附录 C 是词汇表,它指向具体的章节。这可能比用索引查找关键概念更快。



本书的伙伴网站地址:

<http://www.phptr.com/phptrinteractive/>

这个网站的内容非常接近于本书。本书每一章的结尾部分有一系列的思考题,这些题可以使读者实践本章的内容,但是答案在本书中并没有给出,这样可以鼓励读者来到网页上同使用本书的其他人的答案进行比较。并且在这儿可以看到每一章附加的实践题,还伴有提示和指导性的注释。最后在网页上还有一个“作者角”,放了一些不适合本书的附加材料,在这里我们还对本书的问题作出解答,有必要时我们还会粘贴一些本书内容的更正信息。

我们认为本网页是传统出版业同当前异步学习趋势相混合的产物,读者将会发现这里可以提供确实的帮助。

本书所有约定

变量名和语句类型是斜体字。计算机输出的部分用等宽的黑体字。交互式输入和命令的名称用 Courier 字体。在附录 B 中的约定有点不同,黑体字用来显示材料,必须按显示的内容输入。控制码用正常的文本显示。例如,“Control - C”的意思是按住键盘上的“Ctrl”键不放再按字母 C;“Newline”的意思是由于按了“Enter”键而产生的行尾符。

注意:本书中的各行代码,当文本作为单行显示而超出了屏幕的宽度范围时,将会出现错误的行断裂,此时应在第一行的末尾加上字符反斜线(\)。

致 谢

本书是在 Jayne Demsky 的帮助下开始撰写的,他是女王学院(Queens College)的 Prentice - Hall 的销售代表,是他帮助同 Prentice - Hall PTR 的出版人员联系的,谢谢 Jayne。Prentice - Hall 的编辑们给了巨大的帮助,并同丛书的编辑 Mark Taub 以及丛书的顾问 Jeff Gitlin 一起开始工作。Mark 和 Jeff 提出了此套丛书的结构,Ralph Moore 作为编辑使本书格式得到统

一,并在成稿过程中提供了大量的帮助。Nick Radhuber 在生产过程中起到了特殊的作用,感谢他克服了时间的压力才得以使合同如期完成。

Kevin Wall 是本书的技术评论。对保证本书内容的正确性给予了无价的帮助。他对 KornShell 的理解是博大精深的。本书从他的智慧中获益匪浅。也从 Kevin 那儿学习到不少东西,希望在本书印刷前不会出太多的错误。

写本书时用的是 Windows NT。一个窗口运行 Word 字处理程序,另一个窗口运行的是 KornShell 注册到本地计算机上,还有两个窗口运行的 xterm 通过 KornShell 注册到女王学院的 Solaris 和 DEC OSF/1 机器上。但是这儿值得专门提到的是一个窗口连接到 mail 程序上来管理订阅了 U/WIN 的用户以及 U/WIN 开发人员的新的团体。人们通过这些团体可以获得关于 KornShell 特征以及编程技巧的大量的信息资源。David Korn,作为这些团体的积极参与者,同他一起为在本书上工作提供了刺激的学习环境。

众所周知的是作者总是要在致谢的末尾感谢他的家人在写书最困难的时期给予的方便和支持,并且保证这决不是空口的应酬话。非常感谢 Nancy、Josh 和 Alex,他们在本书的酝酿初期给予了很大的支持。

Christopher Vickery
Holliswood, NY

作者简介

Christopher Vickery 是纽约市立大学女王学院计算机科学系的教授,他主要讲授软件设计与开发课程。他的个人主页是:

<http://baggage.cs.qc.edu>.

目 录

第 1 章 设置.....	(1)	实践 2.1 命令构成	(31)
实践 1.1 运行最近的 KornShell	(2)	实践 2.1 命令语法	(31)
实践 1.1 练习	(3)	实践 2.1 练习	(32)
1.1.1 在 Linux 和其他的 UNIX 操作系统	(3)	2.1.1 识别命令名称	(32)
上设置 KornShell	(3)	2.1.2 识别参数	(33)
1.1.2 在 MS Windows 系统上设置	(4)	2.1.3 为程序语法提供文档	(33)
KornShell	(4)	实践 2.1 答案	(34)
实践 1.1 答案	(5)	2.1.1 答案	(34)
1.1.1 答案	(5)	2.1.2 答案	(35)
1.1.2 答案	(5)	2.1.3 答案	(36)
实践 1.1 自测题	(6)	实践 2.1 自测题	(37)
实践 1.2 设置交互式环境	(7)	实践 2.2 命令类型	(39)
实践 1.2 设置 Profile 和环境文件	(7)	实践 2.2 内部命令	(39)
实践 1.2 练习	(7)	实践 2.2 练习	(41)
1.2.1 为程序建立目录	(7)	2.2.1 列出全部内部的命令	(41)
1.2.2 设置 Profile 和环境文件	(8)	2.2.2 使用两个内部工具:print 和 whence	(42)
实践 1.2 答案	(10)	2.2.3 执行外部命令	(42)
1.2.1 答案	(10)	2.2.4 定义别名	(43)
1.2.2 答案	(11)	2.2.5 编写并运行简单的函数	(44)
实践 1.2 自测题	(15)	实践 2.2 答案	(45)
实践 1.3 编辑并运行脚本和函数	(17)	2.2.1 答案	(45)
实践 1.3 练习	(17)	2.2.2 答案	(47)
1.3.1 定义并运行脚本	(17)	2.2.3 答案	(48)
1.3.2 用点(.)执行脚本	(18)	2.2.4 答案	(48)
1.3.3 定义并运行函数	(19)	2.2.5 答案	(50)
实践 1.3 答案	(22)	实践 2.2 自测题	(51)
1.3.1 答案	(22)	实践 2.3 重定向标准输入输出	(53)
1.3.2 答案	(24)	实践 2.3 练习	(53)
1.3.3 答案	(25)	2.3.1 重定向 stdin 和 stdout("<"和">")	(53)
实践 1.3 自测题	(28)	2.3.2 连续重定向控制(>>和>)	(55)
第 1 章 思考题	(29)	2.3.3 重定向 stderr(2>)	(56)
第 2 章 命令语法	(30)		

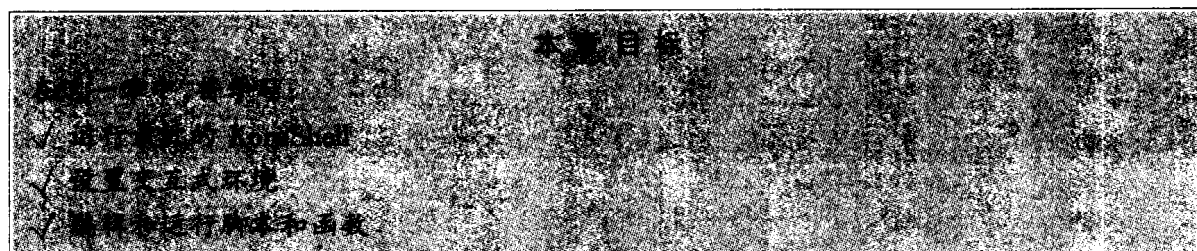
2.3.4 用于重定向文件描述符的连接符号	(57)	实践 4.1 术语	(101)
2.3.5 用于 stdin(<<)的内嵌文档	(58)	实践 4.1 练习	(102)
实践 2.3 答案	(59)	实践 4.1 答案	(103)
2.3.1 答案	(59)	4.1.1 答案	(103)
2.3.2 答案	(63)	实践 4.1 自测题	(104)
2.3.3 答案	(65)	实践 4.2 求值及赋值	(106)
2.3.4 答案	(67)	实践 4.2 练习	(106)
2.3.5 答案	(68)	4.2.1 简单变量求值	(106)
实践 2.3 自测题	(69)	4.2.2 用等号给变量赋值	(107)
第 2 章 思考题	(70)	4.2.3 用 unset 删除变量	(109)
第 3 章 复合命令的语法	(72)	实践 4.2 答案	(111)
实践 3.1 命令序列	(73)	4.2.1 答案	(111)
实践 3.1 练习	(73)	4.2.2 答案	(113)
3.1.1 顺序执行(;)	(73)	4.2.3 答案	(116)
3.1.2 并行执行(&)	(74)	实践 4.2 自测题	(119)
3.1.3 管道()	(75)	实践 4.3 变量引用	(121)
3.1.4 条件命令 && 和 	(77)	实践 4.3 练习	(121)
3.1.5 协作进程(&)	(78)	4.3.1 赋值引用	(121)
实践 3.1 答案	(79)	4.3.2 对变量引用的引用	(122)
3.1.1 答案	(79)	4.3.3 引用单个字符	(123)
3.1.2 答案	(81)	实践 4.3 答案	(124)
3.1.3 答案	(84)	4.3.1 答案	(124)
3.1.4 答案	(88)	4.3.2 答案	(125)
3.1.5 答案	(89)	4.3.3 答案	(127)
实践 3.1 自测题	(90)	实践 4.3 自测题	(128)
实践 3.2 命令组	(92)	第 4 章 思考题	(128)
实践 3.2 练习	(92)	第 5 章 参数	(130)
3.2.1 用{ 和()来给命令分组	(92)	实践 5.1 参数	(131)
3.2.2 用命令组来控制输入/输出重定向	(93)	实践 5.1 练习	(131)
3.2.3 用命令组来控制 && 和 	(94)	5.1.1 验证自己的设置	(131)
实践 3.2 答案	(94)	5.1.2 向函数或脚本传递参数	(133)
3.2.1 答案	(94)	5.1.3 使用特殊参数变量(#@*)	(133)
3.2.2 答案	(95)	5.1.4 使用 set 给位置参数赋值	(134)
3.2.3 答案	(97)	5.1.5 使用 shift 存取位置参数	(135)
实践 3.2 自测题	(98)	实践 5.1 答案	(136)
第 3 章 思考题	(99)	5.1.1 答案	(136)
第 4 章 简单变量	(100)	5.1.2 答案	(138)
		5.1.3 答案	(139)
		5.1.4 答案	(141)

5.1.5 答案	(143)	第 7 章 基本测试	(198)
实践 5.1 自测题	(144)	实践 7.1 if 命令	(199)
实践 5.2 处理选项参数	(146)	实践 7.1 练习	(200)
实践 5.2 练习	(146)	7.1.1 写一个简单的 if、then、...else 语句	(200)
5.2.1 使用 getopt	(146)	7.1.2 使用 if 测试外部命令的输出 ...	(201)
5.2.2 使用 OPTIND 变量	(147)	7.1.3 使用一种可选用方式测试一条命令	(203)
5.2.3 使用 OPTARG 变量	(148)	7.1.4 if 语句嵌套	(203)
5.2.4 定制 getopt 错误信息	(149)	实践 7.1 答案	(204)
实践 5.2 答案	(150)	7.1.1 答案	(204)
5.2.1 答案	(150)	7.1.2 答案	(207)
5.2.2 答案	(151)	7.1.3 答案	(210)
5.2.3 答案	(152)	7.1.4 答案	(210)
5.2.4 答案	(152)	实践 7.1 自测题	(211)
实践 5.2 自测题	(154)	实践 7.2 [[命令...]]操作符	(213)
第 5 章 思考题	(155)	实践 7.2 练习	(213)
第 6 章 高级变量用法	(156)	7.2.1 在 if 语句中使用[[...]]	(213)
实践 6.1 类型变量和范围	(157)	7.2.2 使用[[...]]代替 if	(214)
实践 6.1 练习	(157)	7.2.3 写逻辑表达式	(215)
6.1.1 声明数值型变量和赋值	(157)	实践 7.2 答案	(219)
6.1.2 书写算术表达式	(160)	7.2.1 答案	(219)
6.1.3 使用算术扩展	(162)	7.2.2 答案	(221)
6.1.4 控制变量的范围	(163)	7.2.3 答案	(222)
实践 6.1 答案	(166)	实践 7.2 自测题	(225)
6.1.1 答案	(166)	第 7 章 思考题	(226)
6.1.2 答案	(171)	第 8 章 循环	(227)
6.1.3 答案	(175)	实践 8.1 循环: while 和 until	(228)
6.1.4 答案	(176)	实践 8.1 练习	(228)
实践 6.1 自测题	(180)	8.1.1 使用 while 重复命令	(228)
实践 6.2 数组和复合变量	(182)	8.1.2 使用 break 和 continue 管理一个循环	(230)
实践 6.2 练习	(182)	8.1.3 使用 while 的变体 until	(232)
6.2.1 使用索引数组	(182)	实践 8.1 答案	(233)
6.2.2 声明和使用联合数组	(184)	8.1.1 答案	(233)
6.2.3 使用复合变量和训练函数	(185)	8.1.2 答案	(234)
实践 6.2 答案	(187)	8.1.3 答案	(236)
6.2.1 答案	(187)	实践 8.1 自测题	(237)
6.2.2 答案	(191)		
6.2.3 答案	(193)		
实践 6.2 自测题	(195)		
第 6 章 思考题	(196)		

实践 8.2 循环:列表和算术 for 循环	(239)	10.2.1 用命令的输出作为变量的值 ...	(280)
实践 8.2 练习	(239)	10.2.2 对参数的值使用命令置换	(281)
8.2.1 使用带有值列表的 for 循环	(239)	实践 10.2 答案	(283)
8.2.2 用 for 进行反复计数	(241)	10.2.1 答案	(283)
实践 8.2 答案	(243)	10.2.2 答案	(285)
8.2.1 答案	(243)	实践 10.2 自测题	(287)
8.2.2 答案	(245)	实践 10.3 参数扩展	(289)
实践 8.2 自测题	(247)	实践 10.3 练习	(289)
第 8 章 思考题	(248)	10.3.1 处理默认值和错误	(289)
第 9 章 case 与 select 语句	(249)	10.3.2 区分未设置的变量和空变量 ...	(291)
实践 9.1 case 与 select 语句	(250)	10.3.3 使用字符串和子字符串	(292)
实践 9.1 练习	(251)	实践 10.3 答案	(295)
9.1.1 使用 case 命令	(251)	10.3.1 答案	(295)
9.1.2 使用 select 命令	(252)	10.3.2 答案	(298)
9.1.3 掌握 select 命令的各种用法 ...	(254)	10.3.3 答案	(299)
实践 9.1 答案	(255)	实践 10.3 自测题	(302)
9.1.1 答案	(255)	第 10 章 思考题	(302)
9.1.2 答案	(257)	第 11 章 输入/输出和陷阱处理	(304)
9.1.3 答案	(258)	实践 11.1 READ 和 PRINTF	
实践 9.1 自测题	(259)	命令	(305)
第 9 章 思考题	(261)	实践 11.1 练习	(305)
第 10 章 模式扩展和置换	(262)	11.1.1 使用 read 命令	(305)
实践 10.1 模式	(263)	11.1.2 格式化输出	(308)
实践 10.1 练习	(263)	实践 11.1 答案	(311)
10.1.1 用星号(*)来匹配 0 个或多个		11.1.1 答案	(311)
字符	(263)	11.1.2 答案	(314)
10.1.2 用问号(?)来匹配 1 个字符 ...	(265)	实践 11.1 自测题	(318)
10.1.3 用方括号([])来匹配字符的		实践 11.2 exec 命令	(320)
集合	(266)	实践 11.2 练习	(320)
10.1.4 使用子模式	(267)	11.2.1 用 exec 命令执行另外一个程序	
实践 10.1 答案	(269)		(320)
10.1.1 答案	(269)	11.2.2 用 exec 命令打开和关闭文件	
10.1.2 答案	(273)		(320)
10.1.3 答案	(274)	11.2.3 用 exec 命令读写“感兴趣的”设备	
10.1.4 答案	(276)		(323)
实践 10.1 自测题	(277)	实践 11.2 答案	(324)
实践 10.2 命令置换	(280)	11.2.1 答案	(325)
实践 10.2 练习	(280)	11.2.2 答案	(325)
		11.2.3 答案	(328)

实践 11.2 自测题	(330)	11.3.2 答案	(339)
实践 11.3 截获和产生陷阱	(332)	实践 11.3 自测题	(340)
实践 11.3 练习	(332)	第 11 章 思考题	(341)
11.3.1 截获 INT 和 EXIT 陷阱	(332)	附录 A 自测题答案	(343)
11.3.2 使用 DEBUG 陷阱	(335)	附录 B 内部命令参考	(350)
实践 11.3 答案	(336)	附录 C 术语表	(369)
11.3.1 答案	(336)		

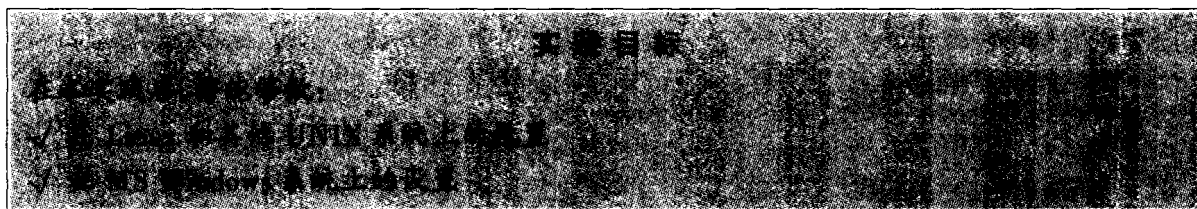
第 1 章 设 置



欢迎使用 KornShell 编程！KornShell 是一种功能强大的计算环境，既可以把它用作命令解释器，又可把它用作编程语言。我们知道，具有位图显示、窗口、图标和鼠标支持的图形用户接口(interface)对依靠计算机做大部分工作的人来说是非常有用的。但是，在需要用实际的计算机代替桌面环境，或者当要自动执行计算任务时，在老式的命令行下不能使用有效的方法操作。

本书中将会有大量的例子，在这里我们假定所使用的计算机已按标准的方法配置好了 KornShell。即使已正在使用 KornShell 也要仔细阅读本章的内容，确保系统能按标准的方法安装操作，本书的其余部也将采用这一方法。如果已经确信当前安装的 KornShell 的版本同本书一致，在学习本书后面部分时将会避免大量的麻烦。

实践 1.1 运行最近的 KornShell



一个活活生生的事实那就是 KornShell 是 AT&T 实验室的 David G. Korn 先生多年来富有创造力的开发成果。更重要的是这个 shell 被移植到越来越多的计算平台上,在今天,事实上可以在任何 UNIX 系统上运行 KornShell,包括 LINUX、Solaris、HP-UX 等其他的 UNIX;同样也可在 Microsoft Windows 9x 和 NT 系统上运行,仅仅写几条 KornShell 的命令就可构成功能强大的程序,并且它能够在所有支持 KornShell 的系统上运行。

但是也有一些不利的因素,那就是它仍有很多老的版本不具备最近版本的一些特征。同样,因为它是一个很有特点的软件环境,在 KornShell 的某些版本中会有一些错误,但是它们都在最近的版本中得到纠正。

下面的内容检查系统上是否有最近版本的 KornShell,并且可以检查出它是否已经运行,否则请升级并运行之。

快速检查

如果有个系统要计划同这本书一起使用,你应该考虑一下它运行的是否是最近版本的 KornShell。现在开始检查,请按 Control-V 键(按住标号为“Ctrl”的键不放再按“V”键)。如果运行的是最近版本的 KornShell,应该能够看到类似于下面的一些信息:

```
Version M-12/28/93f
```

如果当按了 Control-V 键时看到了“^”字符,就有必要在按 Control-V 键前按一下 Esc 键才能看到版本信息。

如果显示的是上面的消息或者是类似于上面消息的结尾字母比 f 大,或者是里面的日期比 12/28/93 晚(值得祝贺的是当前运行的 KornShell 版本同本书是兼容的,并且可以跳过本实验的余下部分),请进入实践 1.2 的学习。然而如果当按 Ctrl-V 时显示的是其他的信息,或者在当前的系统上没有运行 KornShell,就需要找到一个 KornShell 并安装到系统上,如果是一般的用户(不是系统管理员),则要安装到个人的目录上,但是如果是系统管理员,就要安装在系统上为每个用户所共享。

现在如何进行下去要依赖运行的 UNIX 系统或者是 Microsoft Windows。根据所使用的系统来确定应该做练习 1.1.1(UNIX)还是做练习 1.1.2(Windows)。

如果由于某些原因使用的是早期的 KornShell,同样也可以做本书的练习。实际上仅仅有三个主要版本的 KornShell,本书使用的是在 1993 年发布的,称之为 *ksh93*。早一点的版本是 1988 年发布的,称之为 *ksh88*。还有一个在 1988 年前发布的,是原始的版本,但它不属于 AT&T。例如,在 Internet 上能找到 *pdksh*,并且有一个来自 Mortice Kern Systems(MKS)系统

的商业版本,在写这本书的时候,它正在由 Microsoft 授权用在 Windows 2000 上。*pdksh* 和 MKS *ksh* 与 *ksh98* 均不兼容,本书也推荐使用他们。另外在 *ksh88* 下的任何程序在 *ksh93* 下都能够正常运行。Bolsky & Korn 一书的前言部分用了七页的篇幅讲述了由 *ksh88* 升级到 *ksh93* 的特征,但是本书中所使用的大多数功能同 *ksh88* 是兼容的。无论如何,本书也不可能囊括所有的差异。如果不得不使用 *ksh88*,就应该复制一份 Bolsky & Korn 这本书,并仔细查看本书,以便在不能正确工作时看看是不是由于版本问题引起的。



一些老版本的 *ksh* 不能解释 Ctrl-V 字符,如果按了 Ctrl-V 键并没有看到版本信息,可试着用命令 `what/bin/ksh`(如果 KornShell 不在 `/bin/ksh` 下,应该用其所在的路径代替),这下应该能够看到所需要的版本信息了。



在 Solaris 及可能是运行 CDE(公共桌面环境)的其他系统上,按照 `dtksh` 命令,可以找到 *ksh93* 的早期版本。如果没有 *ksh93* 的当前版本,在本书的练习中使用 `dtksh` 也比 *ksh88* 更好。



在 shell 程序中可以使用 `dtksh` 建立图形用户接口(GUI)。如果对做这种软件开发有兴趣,可以查看《Pendergast》一书(前言已提到)。那本书的目的是为有经验的 shell 程序员使用 X Window 系统编程而设计的。如果正在使用本书的程序员不是有经验的 shell 程序员,那一定是有经验的 C/C++ 程序员。

实践 1.1 练习

1.1.1 在 Linux 和其他的 UNIX 操作系统上设置 KornShell

KornShell 是一个可执行的程序,名字就叫 *ksh*,在大多数 UNIX 系统上,包括 Linux 上,均可下载它,请使用下面的 URL:

<http://research.att.com/sw/tools/reuse>

在主页的中间有一个标记着 Binary 的按钮,它的前面有这样一句话“本软件的许可协议适用于非商业范围”,按一下这个按钮将会进入到下一页。在该页中列出了几个软件包,并给出了不同操作系统所对应的软件包的菜单。选择“*ast-base-97*”或者“*ast-base-98*”(根据所使用的操作系统),接受许可协议,下载软件并根据前面所讲安装软件。

还有一个可供选择的方法是在 AIX、HPUX、IRIX、Solaris、SunOS 或 UnixWare 系统上找个 KornShell 的最近版本,可以通过下面的 URL 在全球技术有限公司(Global Technologies Ltd.)的主页上查找:

<http://www.gtllinc.com>

一旦 *ksh* 已经安装在系统上,就可以选择作为默认的 shell 运行,或者选择其他的 shell 作为默认的 shell 然后作为命令运行 *ksh*。

在有些系统中可以采用一个交互式的命令 `chsh`(改变 shell)来改变系统默认的 shell,或者在其他的系统中使用 `passwd` 的一个专门的选项来改变。例如,在 DEC OSF/1 系统上有本地帐号,就可使用 `chsh` 命令改变其默认的 shell;还有如果在 Solaris 系统上有网络帐号,可

以用命令 `passwd -r nisplus -e` 改变。这种命令一方面显示当前 shell 的路径名,另一方面请求输入要改变成默认 shell 的路径名。一旦改变了默认的 shell,注册到系统中时,KornShell 将会开始运行并且解释键入命令行的命令。如果对上述所做没有把握,可以把 KornShell 当作一个单独的命令运行一段时间,直到熟悉了上面的工作。



错误设置默认 shell,会完全拒绝登录系统。如果有一个网络帐号(可以从任何联网的机器上的帐号),不管从哪台计算机上登录,都要确信默认的 shell 路径名的合法性。如果在安装默认的 shell 时出了错,将会使用户根本无法登录。

如果要把 KornShell 作为命令运行,在相应的 shell 提示符下仅仅需要键入 ksh 的路径名。如果在当前 PATH 环境变量中包含着 ksh 所在的目录(在本章后面部分叙述较多),直接键入 ksh 命令即可。

一旦在系统上安装了 ksh,既可以通过作为默认 shell 登录来启动也可以作为命令运行它。按下 Ctrl-V 键。

a) 将会出现什么情况?(在本实践最后有练习答案)

因为使用的是 UNIX 系统,就可跳过下面的练习,当然,除非还想在 Windows 系统中使用 KornShell。

1.1.2 在 MS Windows 系统上设置 KornShell

在 MS Windows 系统(Windows95/98/NT/2000)中使用 KornShell 编程颇有用处,但是如果这样做将会使系统或多或少地类似于 UNIX。有这样几个软件包可供使用,但是其中之一——U/WIN(它代表着“UNIX for Windows”,但是它的发音是“You Win”),它从 David G. Korn 的创造性的开发中获取了大量的有用的东西,由于这个事实,作为该软件包的一部分,它包含着 KornShell 的最终版本特征。

可以从下面的 URL 中下载最近版本的 U/WIN:

<http://research.att.com/sw/tools/uwin>

引用主页上的话“本网站的 U/WIN 二进制文件是作为教育、研究以及评估使用的,U/WIN 的商业许可证可以从 Global Technologies Ltd., Inc 获得”,如果邮寄,还可从 Global Technologies 公司提供的 CD-ROM 中获得代码。(CD 中也包含了大量的 UNIX 的实用程序,可编译运行在运行 U/WIN 的 PC 机上。)可使用下列 URL 与 Global Technologies 公司联系:

<http://www.gtlinc.com>

在 Windows 2000 或 NT 上运行 U/WIN 将会拥有漂亮完美的 UNIX 环境,在 Windows 95/98 下 U/WIN 是非常有用的,但是这些系统不能提供某些必要的特征来支持一些 UNIX 的特征。特别是 Windows 95/98 使用 FAT 文件系统,不支持许多 UNIX 文件系统。同样地,这些系统没有提供足够的多任务环境去支持 UNIX 的一些重要的进程管理。尽管如此,在运行 U/WIN 时,仍然可以使用大多数 KornShell 特征在这些系统下建立程序。