

精通UNIX

Understanding UNIX

〔美〕 Stan Kelly-Bootle 著



電子工業出版社

Publishing House of Electronics Industry

URL:<http://www.phei.co.cn>



Understanding UNIX

精通 Unix

〔美〕 Stan Kelly-Bootle 著

李永峰 曹殿龙 丁红 等译

杨福平 吕 军 校

電子工業出版社

Publishing House of Electronics Industry

内 容 提 要

Unix是非常著名的操作系统，它几乎与现代计算机完全同步地发展壮大着。过去使用PC机的用户常认为：Unix是图形工作站、小型机的系统，我们距离它太遥远了。但随之Internet的推广普及，人们突然间意识到：我必须得学点Unix了！因为稍稍深入到Internet便遇到的是Unix系统界面。本书较全面地介绍了Unix的概貌，基本的Shell、编辑命令、邮件系统以及与Internet的连接方案，远程系统通信等等，并举出了许多Unix的应用实例。它对希望了解和掌握Unix系统的初学者和中高层水平的技术人员会起到极大作用。

本书适合于Unix的初学者，长期使用Internet编程、开发的技术人员。



Copyright©1995 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501.
World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版，SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可，不得以任何形式和手段复制或抄袭本书内容。

书 名：精通Unix

著 者：〔美〕Stan Kelly-Bootle

译 者：李永峰 曹殿龙 丁红 张向阳 张红等

审 校：杨福平 吕军

责任编辑：吕 军

排版制作：北京美迪亚电子信息有限公司

印 刷 者：北京顺义颖华印刷厂

装 订 者：三河赵华装订厂

出版发行：电子工业出版社出版、发行

北京市海淀区万寿路173信箱 邮编：100036 发行部电话：68219077

北京市海淀区万寿路甲15号南小楼三层 邮编：100036 发行部电话：68215345

URL:<http://www.phei.co.cn>

经 销：各地新华书店经销

开 本：787×1092 1/16 印张：23.125 字数：610千字

版 次：1997年4月第1版 1997年4月第1次印刷

印 数：5000册

书 号：ISBN 7-5053-3654-1/TP·1513

定 价：37.00元

著作权合同登记号 图字：01-95-906

凡购买电子工业出版社的图书，如有缺页、倒页、脱页者，本社发行部负责调换
版权所有·翻版必究

序 言

12年以前，Unix主要提供给科研领域的技术人员和科学界的学者们使用。现在，Unix已经是广泛应用在各种计算机上的主流操作系统，有数以百万计的用户在使用着它。

经常看《Unix阅览》杂志“Devil的倡导”专栏的读者都会知道Stan。Stan还写了许多有关计算机其它方面的佳作。象我一样，Stan也是在EDSAC I与II的开拓时代在英国的剑桥大学开始涉足计算机领域的，那是在剑桥的计算机实验室，Maurice Wilkes开创了微程序设计，David Wheeler发明了子程序，因此由Stan写有关Unix的书籍是很合适的。现在许多人已把Unix作为程序设计的开发环境。

第一本Unix书籍出现于80年代初期，并且主要讨论的是技术复杂性。之后出现了许多Unix书籍，它们分别适用于不同的用户。本书的对象是普通计算机用户，它提供了许多其它书籍中没有的知识和经验。

自从70年代初期Unix在Bell实验室问世后，Unix系统增加了许多非常重要的功能，但是基本的系统功能与结构仍保持不变。《精通Unix》为这些基本系统功能提供了简单易学的介绍和说明。在如何通过标准的系统Shell，使用Unix方面还提供了许多颇具实用性的阐述。

Stephen R. Bourne博士

于加利福尼亚的Cupertino

致 谢

我要感谢许许多多编写本书时给予我直接帮助和间接帮助的人们。首先，要感谢SYBEX的员工给予我的随时随地的帮助，他们以各种方式鼓励、激励我完成此书。这其中，我要特别感谢Rudolph Langer博士所给予的鼓励，是他第一个建议我写这本书。

由衷地感谢责任编辑Barbara Gordon和选题策划编辑Dianne King，是他们耐心地帮助我越过了一个又一个难关。

在手稿和其出版物之间，潜藏着数以千计的样式与排版方面的技巧或缺陷，包括错误的字体、标点符号、引号，它们可能会大大增加出错的可能。因此，我非常感谢SYBEX这个编辑小组，包括产品部助理、排版员和美术设计。从输入错误的调整到整洁页面的排布，他们花了很多的时间和精力，从而保证了本书的优美和准确。

我特别感谢如下工作人员：组稿编辑Gary Masters；编辑Odile Sullivan-Tarazi和Armin Brott；项目编辑Kathleen Lattinville和Valerie Potter；作稿编辑Dan Tauber和Aaron Kushner；产品部助理Lisa Haden；产品协调员Sarah Lemas；桌面排版员Thomas Goudie、Len Gilbert和Stephanie Hollier；美术设计Claudia Smelser；以及特技员Guong Le。

我还要感谢那些为Unix贡献其聪明、才智的人们。特别要提到的是，以前我曾与Jim Joyce和Bob Toxen一起开过一次会议，他们对Unix的热情深深地感染了我。我的许多同事，《Unix阅览》和《计算机语言》杂志的许多专栏作家，在过去几年里也给了我许多的帮助，他们是：Steve Bourne、Ken Arnold、Joe Celco、Michael Godfrey、Eric Allman、Nicole Freeman、Tim Parker、Greg Chesson、Bill Plauger、Warren Keuffel、Dave Burnette、Mark Compton、Ken Broadhurst、Dave Chandler、John Dlougosz、Chip Rabinowitch、Lea Anne Bantsari、J. D. Hildebrand，以及默默无闻的Ellipsis。

非常感谢Steve Bourne为本书做了序。

最后，请允许我表达，我对妻子的感激之情！

引 言

Yaub ?

如果Yaub（另一本关于Unix的书籍）需要作一些解释说明的话，毫无疑问！已有许多非常好的文章阐述了Unix的各个方面；其中，有初级的，也有很专业化的。但是，现存的这些作品对新的Unix读者来说不是太浅了，就是太深了。因此，我希望写一本中等水平的、注重实用性的Unix书籍。我认为，读者都极其聪慧并具备极强的求知欲。但这并不能帮助你解决面对着Unix计算机和堆积如山的手册所产生的困惑，也无助于你通过Unix来完成许多有实际意义的工作。过去这几年来，在科学和商业领域中Unix用户的数目大幅度地增长，其应用程序也以罕见的速度增加。与之同步增长的，自然也包括使用该系统所带来的各种问题的数量。

在全世界的教育和科研领域中，Unix早已成为标准的操作系统。计算机学者、物理学家、工程师、生物学家，甚至经济学和社会学界的人士都在使用Unix。最近，计算机技术在人文科学中也扮演了重要角色，而且越来越多地学习历史、考古学、音乐、语言学、文学的学生现在面对的也是Unix中“login:”式的提示符了。除了直接使用计算机的学科外，广泛使用的字处理程序也已经替代了打字机。简单的“Unix易学易用”方面的书籍很难满足广大读者的需要。

在商业领域，Unix正在以惊人的速度增长。现在很多办公室中都在使用Unix系统，这是几年前所无法预见的。所有著名的计算机厂商现在都提供某种版本的Unix，并且许多公司都采用Unix作为其研究、开发及办公系统的计算环境。由于更加快速的Intel和Motorola微处理器的出现，存储器和磁盘的价格越来越低，现在的PC机已经有足够的能力支持各种各样的Unix应用程序，包括各种网络上的应用程度和单独的应用程序。RISC（精简指令集计算机）芯片更进一步帮助拓展了Unix工作站的市场，惠普公司、Sun公司、IBM、Silicon 图象公司及其它厂商都把其工作站产品投向了市场，特别是用在网络服务器、处理复杂的金融模型以及其它的密集型计算机系统之上。

微处理器的革命深刻地改变了Unix世界的面貌。数以千计的中、小型计算机公司，现在使用基于Unix的PC机进行日常的财务管理、电子邮件以及字处理方面的日常工作，在Microsoft与SCO成功地在80286上实现了XENIX之后；接着，便是更为成功的System V版本运行在功能更强的、象Pentium这样的Intel微处理器系统上。所有在DOS和Windows上运行的主要应用程序在Unix下能运行，如电子表格、数据库管理系统、财务软件、字处理程序、桌面排版软件等等。在MIT上开发的X Window系统提供了与机器无关、与网络及硬件无关的图形用户界面（GUI），它比Microsoft的Windows功能更强，也更加漂亮。

与传统大型Unix系统的工作方式不同，这些小系统的安装、管理很少需要专门的系统管理员或Unix专家。随着X Window系统的发展，许多Unix厂商都已简化了Unix管理及操

作的图形界面。你可能听到过Unix是多么神秘且不易掌握之类的传言，不要去理会它！但也不可希望Unix象DOS那样简单。事实上，Unix是一种较为复杂、较难掌握的系统。希望《精通Unix》这本书对你有所帮助。

什么是Unix

象很多有趣的题目一样，Unix不容易准确定义。从底层上看，Unix是操作系统的内核（Kernel），它是1969年Ken Thompson在贝尔实验室设计的一整套监控程序，负责控制、协调多用户计算机系统的工作。内核程序（Kernel）之上是Unix文件系统，负责组织管理数据文件、目录及设备。再上面的一层称为Unix的Shell，它是一个特殊程序，用于解释用户输入的命令，避免命令直接与内核程序打交道。再上面的层，就是自从Unix诞生以来不断增长的Unix工具和实用程序。即使当你把内核程序Kernel、文件系统、Shell以及工具等所有名词放在一起，可能还是想象不出整个系统的概貌。本书的目的就是帮助你更多地学习系统命令和Shell语法。正如我在本书开始时所希望揭示的那样，Unix是一个著名的、成功的范例，也是一个涵有大量历史、社会、经济内容的文化现象——Unix并不是什么神秘的东西。

Unix的可移植性打破了专用软件的统治，使用户摆脱了特定厂商硬件方面的限制。由于70年代和80年代Unix在大学领域广泛流行，它影响了一代的学生和程序设计人员，他们很喜爱这种变革，并且热忠于这个新事物、这个开放式系统的风格。现在，很多那一时代的学生和程序设计人员都处在了很高的职位，他们影响着当前计算机技术的研究和发展方向。

众多的版本

后面将会看到，Unix的宗旨也有消极的一面。在过去一些年里已经出现了许多稍有不同Unix系统分支，有些版本之间的不一致很令人烦躁。与其它领域不一样，在计算机中任何一点，哪怕是最微小的差别，都可能产生灾难性的后果。

在《精通Unix》这本书中，我试图包括所有Unix系统的特性，但也可能会出现本书介绍的情况与你所使用的系统不一样之处。如果你是个Unix的初学者，现在正好是学习System V版本的时候，这个版本囊括了大多数分支，而组成了一个功能很强的标准。

在历经了25年的成熟期后，有些悲观主义者认为Unix已经衰退成了一个单调的、标准化的产品。然而，Unix还在蓬勃发展，Unix社会可以从这个系统的开放策略中得到更多的益处。Unix系统已深入到了如今计算机领域的各个方面。

书中的约定

在本书中使用了几种排版规则，用不同的字型和字体指示不同的含义。

- 标准英文字体表示程序元素、Unix命令、文件名及屏幕输出。例如：termcap、ftp。
- **This font in boldface**指明是用户输入的信息，即所有用户输入用黑体表示。例如，**\$cat poem.1**。

在Shell提示符后输入的命令，例如\$ cat poem.1，则用标准英文字体表示。

实竖线“|”表示管道及逻辑式中的“或”，这相当于许多PC机键盘上的虚竖线。

第一章 准备工作

- 开始使用Unix系统
- 注册
- 命令
- 退出系统
- 选取口令

学习的第一个阶段是认识阶段。Unix是一个庞大、复杂、难以掌握的系统。如果不经认真努力地学习，可能会在某些地方被吓住。如果漫无目的地翻阅堆积如山的文献资料，也可能被众多的命令、选择项，以及过分冗长的细节所吓住。要想成为Unix方面的专家，需从Unix的基础入手。在建立了坚实的基础之后，再按步骤分层次地深入学习，这样会轻松许多。本书包含了大量的实例和练习，以帮助读者掌握所学到的知识，看到学习的成果，而不是纠缠于抽象的书本文字之中。本书还注意介绍事物的产生原因，为什么Unix设计者提供某些特性，并且阐述这些特性是如何实现的。对于内部相关的实例，本书用一些非常简单的方法给予证明，以利于消化吸收实例，而不是孤立地学习。

除非你在计算机领域有很高的水平，否则不能听信Unix是个十分神秘系统的任何传说。可以看到，在世界范围内，数以千计的办公室安装了Unix系统。虽然Unix系统在科学研究领域占统治地位，但如今，在商业领域中，从大型机到PC机，Unix系统均在高速地增长。

怎样开始，你要做什么

无论你使用什么样的计算机硬件系统，在获得使用权并开始使用Unix之前，都有一些先决条件。显然，需要一个合适的终端，它正确地连接在运行Unix的计算机上（最近要注意的是：当前的许多计算机都有能力运行多种操作系统。因此，你要确定正在使用的系统是否正确）。针对不同的硬件和Unix版本，Unix的供应商都会提供精确的安装资料。

当硬件系统连接好，并准备开始时，我们称这种状态为计算机处于联机（online）状态。反之，当由于某种原因丢掉上述连接，则进入脱机（offline）状态，需要重新开始。使用的终端可以是本地的，即通过硬件连线直接连接到计算机上，也可以通过调制解调器和电话线进行远程连接，不同的只是建立的连接方式；一旦进入联机状态，不论是本地连接还是远程连接，Unix的工作方式都是十分相似的。

远程连接是通过Unix系统的拨号功能来完成的，它就象打普通的电话一样简单。在多数情况下，拨号是使用相应的通讯软件自动进行的。实际的操作过程与现场密切相关。因此，你需要请教指导教师或Unix供应商。当然，远程操作会使你偶然面对远程通讯领域中的某些小问题——线路上串音、电子干扰、连接断开等等。同样，如果出现了问题，也会有心理压力。

终端

Unix支持几乎所有现行的各种类型终端，对那些还没有投入市场的终端，稍微改动一下软件便可以使用。对初学者，终端可以看作是一台设备，它带有一个键盘和一个显示器，键盘用于把指令和数据输入到计算机（即输入）；而显示器（也称CRT或监视器）则用来响应输入并显示从计算机来的信息或回答（即输出）。计算机本身通常有一条硬件线路连接的终端，称之为控制台（Console）或系统控制台（System Console）。

终端有哑终端和智能终端之分，但它们的区别界限不十分明显。哑终端拥有一个键盘和显示器，它本身没有处理能力——它的所有处理都是依赖计算机的。智能终端除了具有键盘和显示器外，它本身有内存和处理能力。由于现在微处理器和集成电路的价格低廉，所以真正的哑终端已十分少见。同样由于价格因素，也使许多用户采用PC机作为“智能终端”，它既可联机作为Unix终端，又可脱机作为独立的PC机使用。实际上，超高速微处理器的出现，如Intel 80486、Pentium及Motorola 68030/40，使得现在的微型计算机具备大容量的磁盘和随机存储器RAM。在PC机或工作站上都可以很好地运行Unix，并支持其它终端和PC机。

键盘

无论使用的是哑终端还是智能终端，联机的PC还是运行Unix的PC机，都需要一个键盘，它用于输入而屏幕则作为输出。现在市场上流行着各种各样布局的键盘，键盘上的印字也有各种形式。这使本书介绍指令时，需对某些特殊键盘做相应的调整。对更为复杂的情况，例如某些高智能终端，它们允许人工定义键盘的使用方式，也就是说，可以重新映射键盘，这样就可以改变某些键的作用或键的次序。

重新映射的基本原理是使用模拟软件，它可使一台X牌的终端表现起来就象一台Y牌终端。总之，Unix需要知道使用终端的类型和工作方式，以便它可以正确解释输入的字符流。一般情况下，可打印的字母和数字字符是标准的（通常称为ASCII字符集），但是不可显示的控制字符则根据各种各样的键盘、屏幕、特别是打印机的不同而不同。对这些难以预测的变化，Unix的处理方法是难于把握的。因此，直到你很好地掌握了Unix的各种处理方法之前，最好是让供应商或者系统管理员来设置好你的特殊终端。

为了避免太多的种类，我们将假定读者使用的是标准PC键盘。当遇到非常规的字母、数字键的特殊击键要求时，本书将指出一些最常见的变形。例如，在PC机上称Enter的键，在Unix输入中扮演重要的角色，其它机器的Enter键可以标记为Return、CR、CarRet，或者“**␣**”。

屏幕

显示器与显示适配器的种类变化不会对我们的介绍产生过多影响。除了特别声明的情况以外，我们假定使用的是25行、80列的单色字符终端。当然，Unix支持彩色图形应用程序和其它屏幕显示形式，但是在本书中，基本命令都将使用字符方式。

谁负责什么

在安装大型系统时，有专门指定的人员来管理系统。所以一般情况下，你无需考虑安装终端、设置Unix系统来操作终端等方面。特殊情况出现时，你可以找系统管理员，他会负责用户与Unix系统之间的联络。无论系统管理工作由专职人员承担，还是由工作人员分别承担，都要依据Unix规模的大小以及复杂程度来决定。在小型系统安装中，系统管理员可简单地指派给某个选中的用户，他除了管理系统外还要做其它工作。这个系统管理员可能就是你。不要害怕，本书将解释系统管理员的主要责任。现在假设：已经有人把机器连接好了。

终端上显示的是等待你注册的提示符：

login:

冒号后跟着一个不断闪烁的光标。所谓“提示符”是指一段信息或符号，它指出系统在等待输入。换句话说，系统在提醒你输入字符！光标在屏幕上是一个特殊标记，它指示输入字符将出现的位置。

尽管已经与Unix联机，并且Unix也可以同你会话，但是你仍不是活动用户。必须首先注册，使Unix确认你是一个合法用户，下面是注册的方法。

注册 (Logging In)

在类似DOS的单用户环境中，任何人不必经过许可就可以启动系统。DOS本身并不关心其使用者是谁。如果你想控制DOS系统的使用人员，就必须安装附加的保密系统（机械锁或屏障软件）。对于象Unix这样的多用户系统，必须有严格的保密措施。以防止其它终端上的用户访问你的文件。Unix提供了内置的安全功能，它允许系统管理员控制用户是否可联机注册；系统管理员还可以自由地决定采用哪一级安全机制，自然必须有最小的安全机制限制。

系统管理员为每个用户分配唯一的用户名（也称作注册名）和一个初始口令。系统管理员在系统上记录用户的注册名和口令，并且告诉用户其注册名及口令。一旦你进入系统（将会很快看到），就可以自由地改变你的口令，但改变不了注册名。注册名通常使用的是自己的姓名，用小写字母，如stan或mary。在较大的系统中，有必要使用全名，在系统的安装开始就保证注册名的唯一性。无论采用怎样的方案，注册名必须使用小写字符，并且自始至终，Unix把注册名作为用户的唯一标识符。

口令的选择需要认真考虑，我们将在后面仔细讨论这个问题。现在，要确保每个用户知道其口令，并且只允许用户自身知道其口令。不要忘记自己的口令，否则会给系统管理员增加很多工作，他可能会非常烦恼，发给一个临时的、令人泄气的口令，如dummy。

在login: 提示符后，输入你的注册名。在输入时，可以从屏幕上看到击键的响应。你输入的字符流实际上进入Unix中的一个临时存储区，称之为缓冲区。当你输入字符时，每个字符再送回给屏幕——专业术语称之为回送或显示。大多数键盘的输入都将被回送显示，但你很快会看到非常重要的例外。

输入错误

键盘输入的内容集结在缓冲区中，当按Enter键时，Unix开始检查输入的内容。因此，在按Enter键之前，你还有机会纠正所做的输入。

如果你查觉到有输入错误，那么有两种方法可以补救：抹去并重新输入字符，或者取消输入，然后再重新开始。

按Backspace（退格键）键可以抹去光标左侧的字符。如果键盘没有Backspace键，则可使用Ctrl-H。Ctrl-H的意思是按住Ctrl键的同时按下H键。在有的Unix资料中它的写法为^H。

要删去一整行，可按下Ctrl-U，然后重新开始输入（你的系统可能允许使用#删去前面的字符或使用@删去整个一行。与系统管理员核查或试验一下！）。

回车键（Enter）

当你认为输入的信息正确无误后，按Enter键。在按Enter键之前，Unix会一直等待输入。一般规则是：按Enter键产生一个输入结束的信号，并请求Unix处理你的输入信息。Unix给出一个带有提示符的响应：

Password:

口令是区分字符大小写的。输入的口令要与系统管理员为你设置的口令完全一致，输入时要准确、认真（使用正确的大写和小写字母）；然后按Enter键（这里我最后一次提醒，在每完成一行输入后需按Enter键）。我强调认真二字有两个原因：

- 为了防备你身后的偷看者，口令域中输入的内容不会送到屏幕上。
- 口令可能是（或应该是）奇怪的、非英文单词的、很难的输入组合。

现在Unix同时核对你的注册名和口令。如果注册名合法并且与先前定义的口令相匹配，则注册成功。

注册出错

如果注册出错，就会得到如下信息：

Login incorrect

之后再跟着注册提示符：

Login:

注意Unix在这里做得非常严紧。如果你输入的是stam，而不是stan的话，而随后输入了stan的口令，则Unix不会指出注册名非法；如果你输入的是stan而用mary的口令字，则Unix不会指出口令非法。你得到信息也是注册出错、口令字出错或两者不匹配。扣住这些线索就象是间谍密探游戏的相应部分——它对潜在的偷入者是没有帮助的。如果你得到了Login incorrect信息，检查一下大小写切换键，检查是否用错了大小写字母。如果还不行的话，请

与系统管理员一起进行检查。

注册成功

一旦完成了注册工作，Unix就会根据系统设置按照多种方式进行响应。你或系统管理员都可以定义注册成功后的某种工作环境使之适合你的工作风格。这就是Unix的“双刃剑”：无限灵活性同时也容易出错！在最简单的情况下，注册成功后，迎接你的是一个简单的字符，称作Shell提示符。Bourne Shell程序使用的是\$符号；C Shell程序使用的是%符号（还有其它的Shell提示符，但以上这两种是最常用的）。

在我们进入到更高级的Unix操作之前，Bourne Shell和C Shell提示符的不同对读者是没有影响的。这里的例子使用的是Bourne Shell，如果使用的是C Shell，当你在下面例子中看到\$提示符时，将其想象为%就成了。

本书后面将详细介绍有关Shell的问题。现在我们姑且把它看作成一个特殊的Unix程序，它将解释键盘的输入命令并把解释信息传递到系统变成相应的动作。Unix的Shell有时象DOS中的命令行解释程序COMMAND.COM。现在大多数Unix系统提供多种Shell的选择，允许用户或系统管理员选择适合自己的缺省Shell。缺省这个词经常出现在数据处理中，让我们用简短的时间额外地讨论一下它的意义。

缺省 (Default)

如果你遇到一个简单判断，计算机需要得到一个Yes或No的简单回答，可能90%的时候都回答Yes，例如：

```
Are you sure you want to Exit ? y/n?
```

如果程序已经安排成Enter键作为输入“y”的方式，我们就称y为缺省值。更一般地，在一些选择中，在用户没有输入的情况下，某个选择被程序采纳，则这个选择便是缺省值。在Unix中，你可以随时改变缺省值，或为某个特殊的会话设置新缺省值，对于后面这种情况，在会话之后应恢复原有的缺省值。

欢迎，注册者

除了Shell提示符以外，Unix还可能显示一条或二条信息。

如果Unix无法确定你的终端类型，在它继续处理之前，可能提示如下信息：

```
TERM=(UNKNOWN)
```

或：

```
TERM=(vt100)
```

在第二种情况，vt100（或括号中其它类似的字符）代表了缺省的终端类型。为了通过这一关，需要按Enter键确认这个缺省类型，或者提供实际使用的终端类型。这时你可以寻求系统管理员的帮助。Unix支持的所有终端类型都有一个助记名，vt100（对应于DEC VT100）、wyse 60（对应于Wyse 60）等等。在多数情况下，系统管理员要设置系统，关

闭TERM的信息显示，或者只需简单地按一下Enter键即可。

系统管理员也可以设置日期（在Unix中称作motd），这个信息可以在所有注册用户的终端上显示。类似信息也可以作为一条重要的新闻进行广播（如：The system will be down next Friday），或一条祝贺词（如：Welcome aboard!）。我所用的SCO Unix报告磁盘使用情况信息。你看到的信息可能是：

```
You have mail
```

它提醒你，自从上次清理过邮件信箱（mail box）以后，Unix电子邮件服务又接收到了一条或多条消息。邮件可能来自于本系统上的用户（甚至可能是你发给自己的信件），或者来自于远程系统，该远程系统有你的Unix邮件地址并正确连接在一起。

命令

Shell提示符提示什么呢？回答是Unix命令。这些命令可能有各种各样的形式和长度，从简单的常用命令到极少用、很难记忆的命令。命令本身可能是个相当简单的字符集（但不是所有的命令都带有助记符），而命令可能带有多个选择项或参数。

在下一章中我们将解释这些术语。现在，让我们试用几个简单命令。我们将用黑体表示你输入的内容，一般字体表示回答信息。首先输入命令date，会话形式如下：

```
$ date
Thu Feb 16 09:35:12 PST 1995
$_
```

注意，在日期、时间和时区显示之后，Shell提示符又重新出现，让你输入下一行命令。下一个例子是输入who命令来看看谁在使用本系统。回答将是一个清单，包括所有活动用户名、用户注册的终端、以及用户最近的注册日期和时间。

```
$ who
iwonka tty1a Feb 16 10:05
stan   tty2a Feb 16 15:23
$_
```

Unix允许在同一命令行上输入两条或多条命令。在每条命令之间加上一个分号，在最后一行命令之后按Enter键。

```
$ datd;who
Thu Feb 16 09:35:12 PST 1995
iwonka tty1a Feb16 10:05
stan   tty2a Feb16 15:23
$_
```

下一个例子则显示，如果你输入了一条不存在的命令将会发生什么：

```
$ amelia
amelia: not found
```

```
$ jimmy_hoffa
jimmy_hoffa: not found
$_
```

如果你输入一个Unix不识别的命令，就会得到not found信息，之后接着显示一个Shell提示符。

由于一条错误的输入命令而破坏了计算机系统几乎是不可能的，在一开始便认识到这一点是很重要的。确实是有一些非常危险的命令，如rm或rmdir，如果轻率地使用了它们，可能会把不想删除的文件全部删除了。本书将介绍如何避免这样的灾难，以避免产生不必要的计算机恐惧症。

注销

警告：要注意在每次Unix会话后退出注册。

在介绍了如何在系统中注册并输入了几个简单命令后，现在介绍如何从系统中注销（也称之为退出系统）。虽然看起来是早了点，但事实上，这是一个需尽快掌握的重要命令。如果在有效的注册状态下离开了你的终端，就会丧失所有的保密要点和口令，无论你是在远程终端或本地终端上。任何人都可能趁机获取你的使用权，谁知道其他人是否怀有恶意呢？另外，有些系统还要收取机时或访问费用，或者为连接时间设定配额。退出系统过程依据使用Shell的不同而不同。Bourne Shell退出系统的工作如下：

```
$ exit
```

C Shell需要下面的命令退出系统：

```
% Logout
```

在多数场合，可用Ctrl-D退出系统（在按下Ctrl键的同时按下D键）。系统再次显示Login提示符，表示成功地退出了系统，因此如果你或其他人想要再次成为活动用户，则需要重新注册。如果你是通过调制解调器与系统连接的，则必须按照供应商的指令来挂起。许多Unix当你退出系统时变自动挂断调制解调器连接线，但为避免电话费的不必要增长，应再检查一下。

口令的选择

口令的选择应既便于记忆又不易被其他人猜出。口令应该是有利于记忆，否则你就要把它们写在许多容易取到的地方。它们应该有一定的长度、最少六个字符。你可能听说过很多计算机被窃贼闯入的故事。口令被破译的部分原因是用户使用了显而易见的口令字，如丈夫名、夫人名、宠物名或小孩的名字。SCO Unix和其它版本Unix系统提供了可自动生成口令的程序，它能产生“可发音”但又很模糊口令字，如Klibrugak，你可按音节把它记作“Kli-bru-gak”。另一种有效的诀窍是把字母与数字字符及标点符号、甚至是不显示的控制字符混合使用。

小结

本章介绍了如何注册和退出系统，以及在此期间如何输入一些简单的命令。介绍了如何输入用户名和口令字，以及在使用**Enter**键结束一个输入行之前如何改正输入的错误。

你同时接触到了一些Unix术语，后面将有更详细的介绍。

第二章 简单命令

- Shell介绍
- 内核
- 文件——快速入行
- 命令参数

第一章，已经介绍了如何注册进入Unix，如何使用Date和Who这两个命令。在这一章，将介绍几个相关的简单命令。深入的讨论，将在包括Shell、内核、程序、文件和权限等几个部分中。更深一层的知识将在后面各章里给出。在全面学习Unix的知识之时，对系统有个总体概念是需要的。

有关Shell的更多知识

Shell提示符\$或%，表示一个被你或系统管理员指定的Shell程序正运行在系统时间片上。程序的运行或执行，指的是CPU正按步骤的执行程序指令。其中的有些指令可能会由于各种各样的理由暂时停止运行。例如：Shell程序正常地等待键盘输入或等待你的响应。在你输入命令后，Shell程序会“苏醒过来”，并检查输入的值。

不容易察觉的是，Unix自己可以暂时终止某个程序而把CPU的处理时间分配给其它活动程序。实际上Unix是通过开关控制在用户程序之间切换，从而为多个用户服务。这种切换速度非常的快，以至于每个用户都感觉是自己独占了整个系统。象Unix这样能同时处理多个程序的操作系统称为多任务操作系统。相关术语“多用户”，则表示系统能够支持多个用户终端。所有的多用户系统必须是能够处理多任务的，但是多任务却可能在一个单用户系统上运行。例如PC上的OS/2不是一个多用户系统，但它却能支持单个用户同时运行几个程序。Unix是多用户系统，即它不但是多任务同时也是多用户的系统。

从抽象的术语上讲，“多处理机系统”是一台机器中采用了多几个CPU（中央处理器）的系统。但在当前的系统中，无论是大系统还是小系统都趋向于使用“辅助微处理器”，这些辅助微处理器潜藏在磁盘控制器、图形卡或激光打印机中（有时它们比主处理器功能更强）。这样，“多处理机”的确切含义就有些模糊不清了。这是所有不断进步技术的一个普遍性问题，观念和名词常常跟不上技术的发展。再继续我们刚才的话题，Shell。

Shell是一段很奇特的代码，它在Unix生存期中扮演着非常重要的角色。但它象其它Unix程序一样，也是一个存在于文件中的程序。Bourne Shell是以Stephen R. Bourne博士为主，在1976年编写出来的，当时他就职于AT&T公司的Bell实验室，在Unix开发组里工作。

Bourne Shell不仅是一个命令解释程序也是一个程序设计语言。它拥有变量名、条件分支和所有高级程序设计语言所拥有的语言能力。你能写出自己的Shell程序，例如已知的Shell scripts，它产生一个命令序列清单，这个清单称为Shell文本。Shell解释该文本并运行

它（如果熟悉DOS，你可把Shell看作一个高级的COMMAND.COM命令行解释程序和批处理文件的高级形式）。

如何发现内核……

Unix的心脏是一个主策划程序，称为Kernel（内核）。Unix的内核与计算机外设（磁盘、打印机、终端等等）相互影响，并在约定的条件下分配和调度资源。你不必了解内核的内部机制，它会保护用户程序不受错综复杂的硬件事件影响。当Unix从一个系统移植到另一个系统时，通过对内核的重写——修改内核，使之适应新的CPU，便可使系统能运行在一个新的计算机上。Unix不象DOS那样，被限定必须使用Intel系列的微处理器。Unix能运行在任何适应内核的计算机上。内核程序主要是用C高级程序设计语言（也是由AT&T的Bell实验室开发出来的）编写的。这样做减少了移植Unix到其它系统时的工作量，同时也解释了著名的Unix被普遍采用的原因。

简单地说，内核保护用户程序不受计算机硬件的干扰；Shell则保护用户程序不受内核干扰。当然，你也可以寻求Shell的保护。当前许多的Unix应用程序和用户界面都隐藏Shell和它的提示符，而提供给菜单、窗口和其它选择命令以及运行作业的方法。

文件——快速入门

计算机文件的称谓与它以前被用于存储书信的老式马尼拉文件有点联系。这种老式文件带有记录标签的标题；例如：“Gas Receipts, 1909-1909”，允许原始的随机方式进行存取。目前，你可以把计算机文件简单地看作是磁盘中存储数据和程序的一个场所。磁盘文件具有的记录标签称为文件名，一组文件存储在命名的目录里。在确定文件的位置、存取文件时，你和Unix都可以使用这些目录名和文件名，这将在第三章中介绍。

Unix和大多数其它操作系统提供的主要服务便是：通过文件名处理文件。用户一般不需关心文件的物理位置或结构——Unix保持对这些讨厌细节的追踪。例如用户帐目上可付出的钱款数存放在“第8柱面，从102扇区”开始处（它甚至可以在处理期间移动）。而用户需要知道的只是目录名/usr/payables或其它类似的目录名，以及文件名例如ac.payable。

你（或应用程序）可以存取数据文件，检查或更新它的内容。数据或文本文件一般包括可显示字符和被译成ASCII格式的可打印字符。用户的命令，可以控制从磁盘中读取字符序列、显示或打印文本文件的内容。还有更多的命令允许创建或修改文本文件再存储到磁盘中。你也能拷贝或重新命名文件，或者把一个文件追加到另一个文件的后面。自然的，你能很容易地删除（或杀死）某个文件。

对于多用户系统，还需要提供安全保护功能，以防止其他没有相应权限的用户读写或删除重要的文件。Unix有一个精致的方案，用户和用户组依靠文件的从属关系和权限就能保护他们的文件。系统中的每个文件都带有读、写和执行权限，它们决定了谁能对该文件做什么。

二进制文件包含的程序代码及其格式只能被CPU识别。代码中与ASCII字符相吻合的一些字符（特别是控制字符），在打印或显示一个二进制文件时会显示一些莫名其妙的词句，