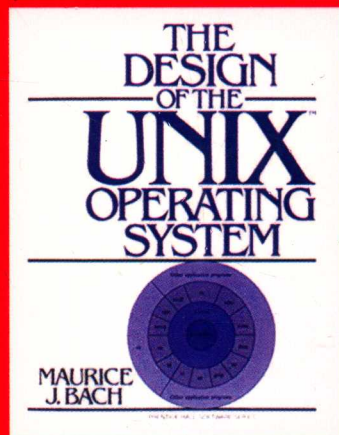


UNIX 操作系统设计

The Design of The UNIX Operating System

[美] 莫里斯·J. 巴赫 (Maurice J. Bach) 著

陈葆钰 王旭 柳纯录 冯雪山 译




Pearson

计算机科学 经典译丛

UNIX 操作系统设计

[美] 莫里斯·J. 巴赫 (Maurice J. Bach) 著

陈葆钰 王旭 柳纯录 冯雪山 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

UNIX操作系统设计 / (美) 莫里斯·J. 巴赫
(Maurice J. Bach) 著; 陈葆钰等译. — 北京: 人民
邮电出版社, 2019. 6
(计算机科学经典译丛)
ISBN 978-7-115-50523-1

I. ①U… II. ①莫… ②陈… III. ①UNIX操作系统—
程序设计 IV. ①TP316.81

中国版本图书馆CIP数据核字(2018)第288178号

版权声明

Authorized translation from the English language edition, entitled THE DESIGN OF THE UNIX[®] OPERATING SYSTEM, 1st Edition, ISBN: 0132017997 by BACH, MAURICE J., published by Pearson Education, Inc. Copyright © 1986 Pearson Education, Inc.

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 Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by POSTS AND TELECOMMUNICATIONS PRESS, Copyright © 2019.

本书中文简体版由 Pearson Education, Inc 授权人民邮电出版社有限公司出版。未经出版者书面许可, 不得以任何方式或任何手段复制和抄袭本书内容。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。
版权所有, 侵权必究。

-
- ◆ 著 [美] 莫里斯·J. 巴赫 (Maurice J. Bach)
译 陈葆钰 王旭 柳纯录 冯雪山
责任编辑 吴晋瑜
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市君旺印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 25
字数: 543千字 2019年6月第1版
印数: 1-2400册 2019年6月河北第1次印刷
- 著作权合同登记号 图字: 01-2017-9040 号

定价: 99.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315
广告经营许可证: 京东工商广登字 20170147 号

内 容 提 要

本书以 UNIX 系统 V 为背景，全面、系统地介绍了 UNIX 操作系统内核的内部数据结构和算法。本书首先对系统内核结构做了简要介绍，然后分章节描述了文件系统、进程调度和存储管理，并在此基础上讨论了 UNIX 系统的高级问题，如驱动程序接口、进程间通信与网络等。

本书可作为大学计算机科学系高年级学生和研究生的教材或参考书，也为从事 UNIX 系统研究与实用程序开发人员提供了一本极有价值的参考资料。

译者序

UNIX 操作系统自 1974 年问世以来，迅速在世界范围内普及。目前，它不但是小型机、高档微型机、工作站系统的主流操作系统，而且已进入中、大型计算机领地，成为“事实上”的标准操作系统，并被国际标准化组织 ISO 等考虑和采纳作为分布式处理系统的本地操作系统参考模型。值得一提的是，20 世纪 90 年代以来，随着运行在 PC 机上的 UNIX 系统的变体——Linux 的出现和迅速普及，UNIX 系统更具生命力。

当前，介绍 UNIX 系统的书籍很多，然而论述 UNIX 系统内部结构的专著却屈指可数。本书是其中非常引人注目的一本。本书作者 Maurice J. Bach 多年来在 AT&T 公司的贝尔实验室工作，对 UNIX 系统的设计思想有深刻了解，又有讲授 UNIX 系统的丰富经验。作者在回顾 UNIX 操作系统的发展演变的基础上，描述了 UNIX 系统 V 内核内部的数据结构和算法，并对其做了深入浅出的分析。在每章之后，本书还给出了大量富有启发性和实际意义的题目。因而，本书不仅可用作大学本科高年级和研究生操作系统课程的教科书和参考书，也为从事 UNIX 操作系统的研究人员或 UNIX 实用程序开发人员提供了极有价值的参考资料。

在本书的翻译过程中，我们尽量保持原著的特色，对书中大量的以 C 伪码形式描述的算法，仍保持 C 语言的结构和格式。因而，阅读本书的读者应具备一定的 C 语言基础。另外，对书中的若干明显错误，我们也一一作了修正。若有错误或不妥之处，恳请指正。

本书是由北京大学计算机系的几位同志合作翻译的：第 1、2、3、4、12 章由柳纯录翻译；第 5、6、8 章由冯雪山翻译；第 7、9 章由王旭翻译；第 10、11、13 章由陈葆珏翻译。全书由陈葆珏修改、定稿。特别值得一提的是，本书的翻译从一开始就得到了杨美清教授的支持和帮助。她还在百忙之中为本书做了校阅，特此表示衷心的感谢。

译者
于北京

前 言

UNIX 系统是由 Ken Thompson 和 Dennis Ritchie 于 1974 年在《ACM 通讯》中的一篇文章中首次提出的 [Thompson 74]。从那时起，UNIX 系统得到迅速传播并在计算机工业中得到广泛采用，越来越多的计算机厂家在他们的机器上提供对 UNIX 系统的支持。UNIX 系统在大学里尤其普遍，它通常被用于操作系统的研究及实例分析。

许多专著和文章曾讨论了系统的各个部分，其中有《贝尔系统技术杂志》1978 年和 1984 年的两个专刊 [BSTJ 78] [BLTJ 84]。还有许多书介绍了 UNIX 系统的用户接口，特别是如何使用电子邮件、如何准备文件及如何使用“shell”的命令解释程序等；《The UNIX Programming Environment》[Kernighan 84]（该书已由机械工业出版社引进出版，中译本名为《UNIX 编程环境》）和《Advanced UNIX Programming》[Rochkind 85] 等书讨论了程序设计环境。本书则着重描述构成操作系统基础（称为内核）的内部算法和数据结构以及它们与程序员接口之间的关系。因此，本书适用于几种环境。首先，它可用作大学高年级本科生或一年级研究生的操作系统课程的教材。使用本书的同时，读者若能参考系统源代码则将获益匪浅，但也可以独立地学习本书。其次，系统程序员可将本书作为参考书，从而能更好地理解内核是如何工作的，并可以将 UNIX 系统中采用的算法与其他操作系统的算法加以比较。最后，UNIX 系统上的程序员能够更深入地了解他们的程序是如何与系统相互作用的，从而编写出更有效、更高级的程序。

本书的内容及组织形式取自我在 AT&T 贝尔实验室在 1983 年和 1984 年期间讲授的一门课程。尽管这门课集中于阅读系统源代码，但我发现，一旦掌握了算法的基本思想，源代码的阅读和理解就会容易得多。在本书中，我已努力使算法的描述尽可能地简单，从而反映出算法所描述的系统的简单性和精巧性。因此，本书并不是用英文逐行地翻译系统，而是描述了各种算法的主要流程，更重要的是，它描述了各种算法是如何相互作用的。算法用类似 C 语言的伪码来表示，从而有助于读者理解自然语言的描述；算法的名字对应于内核内部的过程名。书中的各种插图描绘了系统对各种数据结构进行操作时它们之间的关系。在稍后的一些章节中，我们采用许多小的 C 语言程序来说明一些系统的概念，这些程序的用户是容易明白的。为节省篇幅和清晰，这些例子一般不检查错误条件，而这一点在写程序时是一定要做的。我已经在系统 V 上运行了这些程序；除了某些演示系统 V 的特殊特点的程序外，这些程序也应该能在 UNIX 系统的其他版本上运行。

原来为课程所准备的许多习题已放在每章的最后，它们是本书的重要组成部分。有些习题是直截了当的，用于说明正文中引入的概念。有些习题比较困难，用来帮助读者在一个较深的层次上理解系统。最后，还有些习题具有研究性质，设计这些题目是为了提出问题以供

研究探讨。难度大的题目都标有*号。

本书对 UNIX 系统的描述基于 AT&T 所支持的系统 V，第 2 版，还包括了一些第 3 版的新特点。这是我最熟悉的系统，但我还尽力描述了其他版本对 UNIX 系统的有意义的贡献，特别是 BSD 对系统的修改。本书回避了与特殊的硬件特性有关的问题，力图以通用的术语描述内核硬件的接口，并忽略特定机器的特殊特点。但是，当与机器有关的问题对理解内核的实现十分重要时，本书则讨论得相对详细一些。至少，对这些问题的探讨会突出操作系统中最依赖于机器的部分。

本书的读者必须具有用高级语言进行程序设计的经验，这是理解本书内容的必备条件，最好还有汇编语言的经验。建议读者具有用 UNIX 系统工作的经验，并了解 C 语言 [Kernighan 78]。但是，在编写本书时，我努力使没有这种背景的读者也能理解本书的内容。本书的附录含有系统调用的简单描述，它们足以使读者理解书中的表达方式，但不能作为完整的参考手册。

本书的内容按如下方式组织。第 1 章简要地描述了用户所看到的系统的特点，并给出了系统结构。第 2 章描述了内核结构的一般概貌，并引入一些基本概念。其余的章节按系统结构所表示的组成部分，描述其中各个成分。这些章可分为三部分：文件系统、进程控制和高级问题。本书先讨论文件系统，因为其概念比进程控制容易一些。这样，第 3 章描述了系统缓冲区高速缓存机制，这是文件系统的基础。第 4 章给出文件系统内部使用的一些算法和数据结构。这些算法使用了第 3 章中解释的算法，并讨论了管理用户文件所需要的内部操作。第 5 章说明提供文件系统用户接口的系统调用，这些系统调用使用了第 4 章的算法来存取用户文件。

第 6 章转向进程控制，其中定义了进程的上下文，讨论了控制进程上下文的内部内核原语。特别地讨论了系统调用接口，中断处理及上下文切换。第 7 章给出了控制进程上下文的系统调用。第 8 章讨论了进程调度问题。第 9 章的内容是存储管理，其中包括对换和请求调页系统。

第 10 章讨论了通用驱动程序接口，特别讨论了磁盘驱动程序和终端驱动程序。尽管从逻辑上说设备是文件系统的一部分，但是，因为进程控制的问题要在终端驱动程序中出现，所以，对设备的讨论推迟到这一章。这一章也是通向本书其余章节中所给出的更高级的问题的桥梁。第 11 章讨论进程间通信和网络问题，其中包括系统 V 的消息、共享存储区及信号量，还有 BSD 的套接字。第 12 章解释了紧密耦合的多处理机 UNIX 系统。第 13 章研究了松散耦合的分布式系统。

前 9 章的内容可以在一学期的操作系统课程中完成。其余各章的内容可以在高级讨论班中进行讨论，并同时作各种课题研究。

至此，本人要作几点说明。可以确切地说，本书没有作系统性能方面的讨论，也没有提出任何用于系统安装的配置参数。这些数据会因机器类型、硬件配置、系统版本和实现以及应用类型等的不同而不同。同时，我有意地尽量避免预测 UNIX 操作系统的未来发展。所讨论的高级问题并不意味着 AT&T 就要提供这些特别的特性，甚至也不意味着那些特殊的领域正在开发研究中。

资源与支持

本书由异步社区出品，社区 (<https://www.epubit.com/>) 为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书源代码；
- 书中彩图文件。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the latter being the active tab. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Page Position (Line Number)), and '勘误印次:' (勘误次数). Below these fields is a large text area for entering the勘误 details, with a rich text editor toolbar above it containing icons for bold, italic, underline, list, link, and other text formatting options. At the bottom right of the form, there is a '字数统计' (Character Count) label and a '提交' (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc.



异步社区



微信服务号

目 录

第 1 章 系统概貌	1	3.7 习题	47
1.1 历史	1	第 4 章 文件的内部表示	48
1.2 系统结构	3	4.1 索引节点	48
1.3 用户看法	4	4.1.1 定义	48
1.3.1 文件系统	4	4.1.2 对索引节点的存取	50
1.3.2 处理环境	8	4.1.3 释放索引节点	53
1.3.3 构件原语	10	4.2 正规文件的结构	53
1.4 操作系统服务	11	4.3 目录	58
1.5 关于硬件的假设	12	4.4 路径名到索引节点的转换	59
1.5.1 中断与例外	13	4.5 超级块	61
1.5.2 处理机执行级	13	4.6 为新文件分配索引节点	62
1.5.3 存储管理	14	4.7 磁盘块的分配	68
1.6 本章小结	14	4.8 其他文件类型	71
第 2 章 内核导言	15	4.9 本章小结	71
2.1 UNIX 操作系统的体系结构	15	4.10 习题	72
2.2 系统概念介绍	17	第 5 章 文件系统的系统调用	74
2.2.1 文件子系统概貌	17	5.1 系统调用 open	75
2.2.2 进程	19	5.2 系统调用 read	78
2.3 内核数据结构	27	5.3 系统调用 write	82
2.4 系统管理	28	5.4 文件和记录的上锁	83
2.5 本章小结	28	5.5 文件的输入/输出位置的 调整——lseek	83
2.6 习题	29	5.6 系统调用 close	84
第 3 章 数据缓冲区高速缓冲	30	5.7 文件的建立	86
3.1 缓冲头部	30	5.8 特殊文件的建立	87
3.2 缓冲池的结构	32	5.9 改变目录及根	88
3.3 缓冲区的检索	33	5.10 改变所有者及许可权方式	90
3.4 读磁盘块与写磁盘块	43	5.11 系统调用 stat 和 fstat	90
3.5 高速缓冲的优点与缺点	45	5.12 管道	90
3.6 本章小结	46		

5.12.1 系统调用 pipe	91	6.5 进程地址空间的管理	141
5.12.2 有名管道的打开	92	6.5.1 区的上锁和解锁	142
5.12.3 管道的读和写	92	6.5.2 区的分配	142
5.12.4 管道的关闭	94	6.5.3 区附接到进程	143
5.12.5 举例	95	6.5.4 区大小的改变	144
5.13 系统调用 dup	95	6.5.5 区的装入	146
5.14 文件系统的安装和拆卸	97	6.5.6 区的释放	148
5.14.1 在文件路径名中跨越 安装点	100	6.5.7 区与进程的断接	148
5.14.2 文件系统的拆卸	103	6.5.8 区的复制	150
5.15 系统调用 link	104	6.6 睡眠	151
5.16 系统调用 unlink	108	6.6.1 睡眠事件及其地址	151
5.16.1 文件系统的一致性	109	6.6.2 算法 sleep 和 wakeup	152
5.16.2 竞争条件	110	6.7 本章小结	156
5.17 文件系统的抽象	113	6.8 习题	156
5.18 文件系统维护	114	第 7 章 进程控制	159
5.19 本章小结	116	7.1 进程的创建	159
5.20 习题	116	7.2 软中断信号	166
第 6 章 进程结构	121	7.2.1 软中断信号的处理	169
6.1 进程的状态和状态的转换	121	7.2.2 进程组	175
6.2 系统存储方案	125	7.2.3 从进程发送软中断信号	175
6.2.1 区	125	7.3 进程的终止	176
6.2.2 页和页表	126	7.4 等待进程的终止	178
6.2.3 内核的安排	128	7.5 对其他程序的引用	181
6.2.4 u 区	130	7.6 进程的用户标识号	188
6.3 进程的上下文	131	7.7 改变进程的大小	191
6.4 进程上下文的保存	133	7.8 shell 程序	193
6.4.1 中断和例外	133	7.9 系统自举和进程 init	195
6.4.2 系统调用的接口	136	7.10 本章小结	198
6.4.3 上下文切换	139	7.11 习题	199
6.4.4 为废弃返回 (abortive return) 而保存上下文	140	第 8 章 进程调度和时间	206
6.4.5 在系统和用户地址空间之间 拷贝数据	141	8.1 进程调度	206
		8.1.1 算法	206
		8.1.2 调度参数	207
		8.1.3 进程调度的例子	210

8.1.4 进程优先权的控制	212	10.3.1 字符表 clist	273
8.1.5 公平共享调度	213	10.3.2 标准方式下的终端驱动程序	276
8.1.6 实时处理	214	10.3.3 原始方式下的终端驱动程序	280
8.2 有关时间的系统调用	215	10.3.4 终端探询	282
8.3 时钟	218	10.3.5 建立控制终端	283
8.3.1 重新启动时钟	218	10.3.6 间接终端驱动程序	283
8.3.2 系统的内部定时	218	10.3.7 注册到系统	284
8.3.3 直方图分析	220	10.4 流	285
8.3.4 记账和统计	223	10.4.1 流的详细的示例	288
8.3.5 计时	223	10.4.2 对流的分析	290
8.4 本章小结	224	10.5 本章小结	291
8.5 习题	224	10.6 习题	291
第 9 章 存储管理策略	226	第 11 章 进程间通信	294
9.1 对换	226	11.1 进程跟踪	294
9.1.1 对换空间的分配	226	11.2 系统 V IPC	297
9.1.2 进程的换出	229	11.2.1 消息	299
9.1.3 进程的换入	232	11.2.2 共享存储区	304
9.2 请求调页	236	11.2.3 信号量	308
9.2.1 请求调页的数据结构	238	11.2.4 总的评价	316
9.2.2 偷页进程	243	11.3 网络通信	317
9.2.3 页面错	246	11.4 套接字	318
9.2.4 在简单硬件支持下的请求调页系统	253	11.5 本章小结	322
9.3 对换和请求调页的混合系统	254	11.6 习题	323
9.4 本章小结	254	第 12 章 多处理机系统	325
9.5 习题	255	12.1 多处理机系统的问题	325
第 10 章 输入/输出子系统	258	12.2 主从处理机解决方法	326
10.1 驱动程序接口	258	12.3 信号量解决方法	328
10.1.1 系统配置	259	12.3.1 信号量定义	329
10.1.2 系统调用与驱动程序接口	261	12.3.2 信号置实现	329
10.1.3 中断处理程序	267	12.3.3 几个算法	336
10.2 磁盘驱动程序	268	12.4 Tunis 系统	340
10.3 终端驱动程序	271	12.5 性能局限性	340

12.6 习题	340	模型	356
第 13 章 分布式 UNIX 系统	342	13.5 本章小结	357
13.1 卫星处理机系统	343	13.6 习题	358
13.2 纽卡斯尔连接	350	附录 系统调用	361
13.3 透明型分布式文件系统	353	索引	377
13.4 无存根进程的透明分布式		参考资料	382

第1章 系统概貌

UNIX 系统自从 1969 年问世以来已经变得相当流行，它运行在从微处理机到大型机的具有不同处理能力的机器上，并在这些机器上提供公共的执行环境。UNIX 系统可分成两部分，第一部分由一些程序和服务组成，其中包括 shell 程序、邮件程序、正文处理程序包及源代码控制系统等，正是这些程序和服务使得 UNIX 系统环境如此受欢迎。它们是用户立即可见的部分。第二部分由支持这些程序和服务的操作系统组成。本书给出了该操作系统的详细描述，它着重描述由美国电话电报公司 (AT&T) 生产的 UNIX 系统 V，但也考虑了其他版本所提供的颇有意义的特征。本书考查了在该操作系统中使用的主要数据结构和算法，而这些数据结构和算法最终向用户提供了标准用户界面。

本章是 UNIX 系统的引言，它回顾了 UNIX 系统的历史并勾画出了整个系统结构的轮廓。下一章将对该操作系统做更详细的介绍。

1.1 历史

1965 年，贝尔实验室和通用电气公司及麻省理工学院的 MAC 课题组一起联合开发一个被称为 Multics [Organick 72] 的新操作系统。Multics 系统的目标是要向大的用户团体提供对计算的同时访问，支持强大的计算能力与数据存储，以及允许用户在需要的时候容易地共享他们的数据。贝尔实验室中后来参加 UNIX 系统早期开发的许多人当时都参加了 Multics 工作。虽然 Multics 系统的原始版本于 1969 年在 GE645 计算机上运行了，但它既没能提供预定的综合计算服务，也不清楚自己的开发目标。结果，贝尔实验室退出了这一项目。

在结束了 Multics 工程上的工作时，贝尔实验室计算科学研究中心的成员们处于缺乏“方便的交互式计算服务”的境况之中 [Ritchie 84a]。为了改善他们的程序设计环境，Ken Thompson、Dennis Ritchie 及其他人勾画出了一个纸面上的文件系统设计——它后来就演化为 UNIX 文件系统的早期版本。Thompson 编写了若干程序，模拟所建议的文件系统行为，以及模拟在请求调页环境下程序的行为。他甚至为 GE645 计算机的简单内核进行了编码。与此同时，他用 Fortran 语言为 GECOS 系统 (Honeywell635) 编写了名为“宇宙旅行”的游戏程序。但这个程序是不能令人满意的，因为它很难控制“宇宙飞船”，并且该程序运行开销太大。Thompson 后来发现几乎无人问津的 PDP-7 计算机能提供很好的图形显示和廉价的执行开销。为 PDP-7 开发“宇宙旅行”程序使 Thompson 了解到了关于该机器的细节，但是它的程序开发环境要求先在 GECOS 机上进行程序的交叉汇编，而后把纸带带到 PDP-7 上输入。为了创建一个较好的开发环境，Thompson 和 Ritchie 在 PDP-7 上实现了他们的系统设计，其中包括 UNIX

文件系统、进程子系统的早期版本及少量实用程序。终于，新系统再也不需要把 GECOS 系统作为开发环境，而能够自我支持了。这个新的系统被命名为 UNIX。UNIX 是针对 Multics 的双关语，它是计算科学研究中心的另一名成员 Brian Kernighan 想出来的。

虽然 UNIX 系统的早期版本是大有前途的，但直到被用于实际项目之前它并没能发挥出自身的潜力。因此，为给贝尔实验室的专利部门提供一个正文处理系统，1971 年 UNIX 系统被移植到 PDP-11 上。该系统的特征是规模小：内存中 16KB 用于系统，8KB 用于用户程序；磁盘 512KB，每个文件限定长度为 64KB。在它初次成功之后，Thompson 开始动手为这个系统实现 Fortran 编译程序。但是在 BCPL [Richards 69] 的影响下开发出来的却是 B 语言。B 语言是解释性语言，在性能上有所退步——这是这类语言的共同特征。因此，Ritchie 把 B 语言发展成他称之为 C 的语言，C 语言允许产生机器代码、说明数据类型及定义数据结构。1973 年，他用 C 语言重写了 UNIX 操作系统。这一步在当时并不太引人注目，但对 UNIX 的外部用户接受它却具有极大的影响。这之后，贝尔实验室的装机数目增加到 25 个，并且形成了一个 UNIX 系统小组，以提供内部支持。

由于美国电话电报公司 1965 年与联邦政府签署了反垄断法，所以这时它不能销售计算机产品。但是，美国电话电报公司把 UNIX 系统提供给了那些请求把 UNIX 系统用于教育目的的大学。该公司信守了反垄断法的条款，它既没有为 UNIX 系统做广告，也没有销售和支持 UNIX 系统。然而，UNIX 系统的声望却在稳步增长。1974 年，Thompson 与 Ritchie 在《ACM 通讯》上发表了一篇描述 UNIX 系统的文章 [Thompson 74]，进一步促进了它的可接受性。到 1977 年，UNIX 系统的装机数目已经增长到大约 500 个，其中有 125 个在大学。UNIX 系统开始在业务电话公司流行起来，为程序开发、网络事务操作服务及实时服务（通过 MERT [Lycklama78a]）提供了良好的环境。这时，UNIX 系统的许可证被提供给商业机构，同时也向大学提供。1977 年，交互系统公司（Interactive Systems Corporation）成了 UNIX 系统的第一个增值转卖商（VAR）^①，他们增强了它，使之在办公室自动化环境中使用。同样，1977 年也是标志 UNIX 系统首次被“移植”到非 PDP 机（即稍加改变或完全不变而在其他机种上运行）——Interdata 8/32 上的一年。

随着微处理机的日益普及，其他公司也把 UNIX 系统移植到新的机器上，但是它的简单清晰的特点吸引着很多开发者以他们自己的方式增强 UNIX 系统，结果在基本系统上产生若干变体。从 1977 年到 1982 年这一时期，贝尔实验室把若干 AT&T 变体综合成一个单个系统，这就是大家都知道的商用 UNIX 系统Ⅲ。随后，贝尔实验室又把若干特征加到系统Ⅲ上，称为新产品 UNIX 系统 V^②，1983 年 1 月，美国电话电报公司发布它对系统 V 的正式支持。然而，加利福尼亚大学伯克利分校的人们已经开发了一个 UNIX 系统的变体，它的最新版本称为 4.3BSD (Berkeley Software Distribution)，是配在 VAX 机上的，它提供了一些新的有意义

① 增值转卖商把具体应用加到计算机系统中以满足特定的市场需要。他们销售的是应用而不是销售这些应用赖以运行的操作系统。

② 对系统Ⅳ发生了什么？事实上，系统Ⅳ是 UNIX 系统的一个内部版本，它被融入系统 V 中去了。

的特征。本书将着重描述 UNIX 系统 V，但也偶尔谈及 BSD 系统所提供的那些特征。

到 1984 年年初，世界上大约安装了 100 000 个 UNIX 系统，它们运行在从微处理机到大型机的具有显著差异的计算能力的机器上，运行在出自不同的制造厂家的生产线的机器上。没有任何其他操作系统能与之匹敌。UNIX 系统的普及与成功可归结为如下一些原因。

- 该系统以高级语言编写，使之易读、易懂、易修改、易移植到其他机器上。据 Ritchie 估计，用 C 语言编写的第一个系统与用汇编语言书写的系统相比，要大而且慢 20%~40%。但是，采用高级语言的优点要远远超过它的缺点（见 [Ritchie 786] 第 1965 页）。
- 它有一个简单的用户界面但具有提供用户所希望的服务的能力。
- 它提供能够由较简单的程序构造出复杂程序的原语。
- 它使用了在维护上是容易的、在实现上是高效的层次式文件系统。
- 文件采用字节流这样的一致格式，使应用程序易于编写。
- 它为外围设备提供了简单一致的接口。
- 它是一个多用户、多进程系统，每个用户都能同时执行几个进程。
- 它向用户隐蔽了机器的体系结构，使用户易于编写在不同硬件实现上运行的程序。

简单性与一致性突出了 UNIX 系统的宗旨，上面列出的大部分原因都讲的是简单性与一致性。

虽然操作系统和很多命令程序是用 C 语言编写的，但是 UNIX 系统支持其他语言，包括 Fortran、Basic、Pascal、Ada、Cobol、Lisp 及 Prolog 等。UNIX 系统能支持具有编译程序或解释程序的任何语言；UNIX 系统还能支持一个系统接口，该接口把用户对操作系统服务的请求映射到 UNIX 系统使用的一组标准请求上。

1.2 系统结构

图 1-1 绘出了 UNIX 系统的高层次的体系结构。该图中心的硬件部分向操作系统提供的基本服务将在 1.5 节中描述。操作系统直接^①与硬件交互，向程序提供公共服务，并使它们同硬件特性隔离。当我们把整个系统看成层的集合时，操作系统通常称为系统内核，或简称内核 (kernel)，此时强调的是它同用户程序的隔离。因为程序是不依赖于其下面的硬件的，所以，如果程序对硬件没有做什么假定的话，就容易把它们在不同硬件上运行的 UNIX 系统之间搬动。比如，那些假定了机器字长的程序比起没假定机器字长的程序来说，就较难于搬到其他机器上。

^① 在 UNIX 系统的某些实现中，该操作系统同内核 (native operating system) 接口，再由核与其下面的硬件接口，并向 UNIX 系统提供必要的服务。这样的配置可以使其他操作系统及其应用程序能够同 UNIX 系统平行地运行在装置上。这样的一种配置的经典例子是 MERT 系统 [Lycklama 78a]。更近期的配置包括面向 IBM 系统/370 计算机 [Felton 84] 及面向 UNIVAC1100 系列计算机 [Bodenstab 84] 的实现。

外层的程序，诸如 shell 及编辑程序 (ed 与 vi)，是通过引用一组明确定义的系统调用而与内核交互的。这些系统调用通知内核为调用程序做各种操作，并在内核与调用程序之间交换数据。图中出现的一些程序属于标准的系统配置，就是大家知道的命令，但是，由名为 a.out 的程序所指示的用户私用程序也可以存在于这一层。此处的 a.out 是被 C 编译程序产生的可执行文件的标准名字。其他应用程序能在较低层的程序之上构建，因此它们存在于本图的最外层。比如，标准的 C 编译程序 cc 就处在本图的最外层：它调用 C 预处理程序、两遍编译程序、汇编程序及装入程序（称为连接-编辑程序），这些都是彼此分开的低层程序。虽然该图对应用程序只描绘了两个级别的层次，但用户能够对层次进行扩充，直到级别的数目适合于自己的需要。确实，为 UNIX 系统所偏爱的程序设计风格鼓励把现存程序组合起来去完成一个任务。

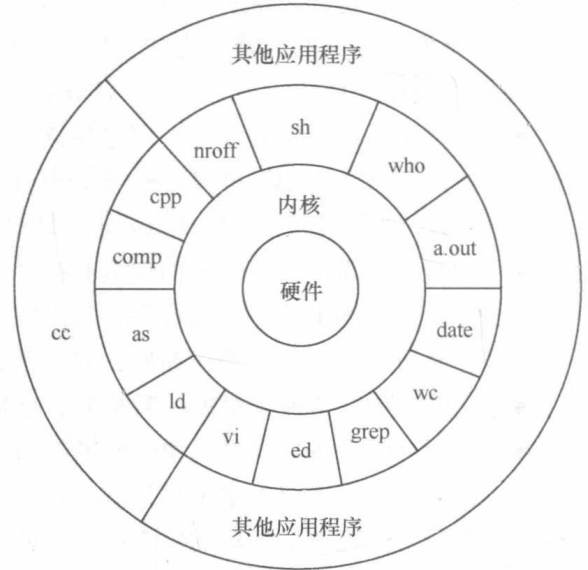


图 1-1 UNIX 系统的体系结构

一大批提供了对系统的高级看法的应用子系统及应用程序，诸如 shell、编辑程序、SCCS (source code control system) 及文档准备程序包等，都逐渐变成了“UNIX 系统”这一名称的同义语。然而，它们最终都使用由内核提供的低层服务，并通过系统调用的集合来利用这些服务。系统 V 中大约有 64 个系统调用，其中将近 32 个是常用的。它们有简单的可选项，这些可选项使系统调用容易使用，但是却向用户提供了很多能力。系统调用的集合及实现系统调用的内部算法形成了内核的主体，因而本书所要介绍的对 UNIX 操作系统的研究，就变成了对系统调用及其相互作用的详细研究和分析。简言之，内核提供了 UNIX 系统全部应用程序所依赖的服务，它也定义了这些服务。本书将频繁使用“UNIX 系统”“内核”或“系统”等术语，但其含义是指 UNIX 操作系统的内核，并且在上下文中是清楚的。

1.3 用户看法

本节简要地介绍 UNIX 系统的高层特征，诸如文件系统、处理环境及构件原语（比如，管道），后面几章将详细地探讨内核对这些特征的支持。

1.3.1 文件系统

UNIX 文件系统有如下特点。