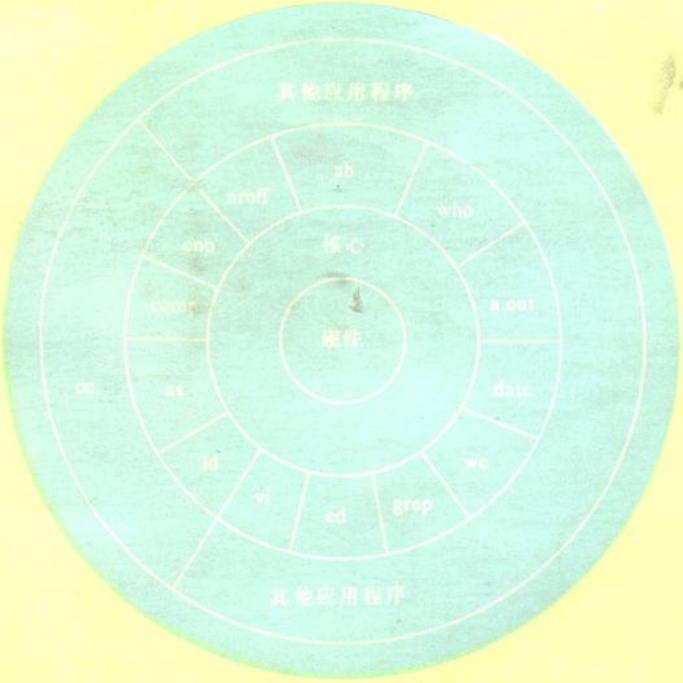
UNIX
操作系统设计

〔美〕 莫里斯·贝奇 著



北京大学出版社

UNIX 操作系统设计

[美] 莫里斯·贝奇 著
陈葆珏 王 旭 柳纯录 冯雪山 译

北京大学出版社

内 容 简 介

本文以 UNIX 系统 V 为背景，全面地、系统地介绍了 UNIX 操作系统核心的内部的数据结构和算法。全书共十三章，按以下几部分组织：第一部分（第一到二章）描述系统的一般概貌；第二部分（第三到五章）介绍文件系统；第三部分（第六到九章）介绍进程调度和存储管理；第四部分（第十到十三章）讨论了若干 UNIX 系统的高级问题，例如进程间通信和网络问题。

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教学参考书，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。

Maurice J. Bach

The Design of The UNIX™ Operating System

Prentice-Hall, Inc., 1986

UNIX 操作系统设计

[美] 莫里斯·贝奇 著

陈葆珏 王 旭 柳纯录 冯雪岗 译

责任编辑：彭令华

*

北京大学出版社出版

（北京大学校内）

北京雪花日历印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

850×1168 毫米 32 开本 18 印张 456 千字

1989 年 11 月第一版 1989 年 11 月第一次印刷

印数：0001—4,000 册

ISBN 7-301-00991-7/TP·027

定价：9.40 元

译者序

UNIX 操作系统自 1974 年问世以来，迅速地在世界范围内推广。目前，它不仅是小型机、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，并正在被国际标准化组织 ISO 等考虑和采纳作为分布处理系统的本地操作系统参考模型。

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。本书是其中最引人注目的一本。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 核心内部的数据结构和算法，并对其作了深入浅出的分析。在每章之后，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。

在翻译本书过程中，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。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：第一、二、三、四、十二章由柳纯录同志翻译；第五、六、八章由冯雪山同志翻译；第七、九章由王旭同志翻译；第十、十一、十三章由陈葆珏同志翻译。全书由陈葆珏同志修改、定稿。特别值得一提的是，本书的翻译从一开始就得到了杨芙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。

序　言

UNIX 系统是由 Ken Thompson 和 Dennis Ritchie 在 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 系统已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”在 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、或如何使用称为“shell”的命令解释程序等；“The UNIX Programming Environment”[Kernighan 84]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(称为核心)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，从而能更好地理解核心是如何工作的，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。在本书里，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；它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；算法的名字对应于核心内部的过程名。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。在稍后的一些章中，采用了许多小的 C 语言程序来说明一些系统的概念，这些程序用户是容易明白的。为节省篇幅和清晰起见，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 的特殊特点的程序以外，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。有些习题是直接了当的，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核心硬件的接口，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核心的实现十分重要时，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，在编写此书时，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，但并不作为完整的参考手册。

本书的内容按如下方式组织。第一章，系统概貌，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。第二章描述了核心结构的一般概貌，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，第三章描述了系统缓冲区高速缓存机制，这是文件系统的基础。第四章给出文件系统内部使用的一些算法和数据结构。这些算法使用了第三章中解释的算法，并讨论了管理用户文件所需要的内务操作。第五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四章的算法来存取用户文件。

第六章转向进程控制，其中定义了进程的上下文，讨论了控制进程上下文的内部核心原语。特别地讨论了系统调用接口，中断处理及上下文切换。第七章给出了控制进程上下文的系统调用。第八章讨论了进程调度问题。第九章的内容是存储管理，其中包括对换和请求调页系统。

第十章讨论了通用驱动程序接口，特别地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。第十一章讨论进程间通讯和网络问题，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。第十二章解释了紧密耦合的多处理机 UNIX 系统。第十三章研究了松散耦合的分布式系统。

前九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。

我很高兴在这里感谢许多朋友和同事在我写书期间给予的帮助。他们鼓励我，并对手稿提供了富有建设性的建议。我深深感谢 Ian Johnstone，是他建议我写这本书，在开始阶段给我勇气，并审阅了头几章的最初草稿。Ian 还教给我许多写书的技巧，对此我永远感激不尽。Doris Ryan 从一开始就在鼓励帮助我，她富于思想并乐于助人。我永远感谢她的帮助。

Dennis Ritchie 解答了许多 UNIX 系统的历史和技术背景问题。还有许多人自愿地花时间和精力审阅手稿，我应大大感谢他们的详细建议。他们是 Debby Bach，Doug Bayer，Lenny Brandwein，Steve Buroff，Tom Butler，Ron Gomes，Mesut Gunduc，Laura Israel，Dean Jagels，Keith Kelleman，Brian Kernighan，Bob Martin，Bob Mitze，Dave Nowitz，Michael Poppers，Marilyn Safran，Curt Schimmel，Zvi Spitz，Tom Vaden，Bill Weber，Larry Wehr 和 Bob Zarrow。Mary Fruhstuck 在准备手稿的排版中给予了热情的帮助。我要感谢我的上司，他们对这个项目自始至终都给予支持；还要感谢我的同事们，在 AT&T 贝尔实验室，他们提供了如此鼓舞人的气氛和优秀的工作环境。John Wait 及 Prentice-Hall 的工作人员提供了许多有价值的帮助，使得本书最终成为目前的形式。最后，但并非最不重要，我感谢我的妻子，Debby，她给了我许多感情上的支持，没有她，我永远不会成功。

目　录

第一章　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、正文处理程序包以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。它考察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，贝尔电话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机的同时访问，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，而且，连它自己也不清楚究竟什么时刻算达到开发目标了。结果，贝尔实验室退出了这一项目。

在他们结束了 Multics 工程上的工作的时候，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 计算机的简单核心进行了编码。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 后来发现了一个几乎无人问津的 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 学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，而能够自己支持自己了。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，但直到它用于实际项目之前它并没能发挥出它的潜力。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。该系统的特征是它的规模小：内存中 16K 字节用于系统，8K 字节用于用户程序；磁盘 512K 字节，每个文件限定长度为 64K 字节。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，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，但它对其外部用户的可接受性却具有极大的影响。这期间，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系统提供给了请求把UNIX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 [Lycklama 78a])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，结果导致在基本系统上的若干变体。

① 增值转卖商把具体应用加到计算机系统上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。