****

**THE LINUX PROGRAMMING INTERFACE**

**A Linux and UNIX System Programming Handbook**

**MICHAEL KERRISK**

no starch press

San Francisco

*Translated by: Kevin*

***本资料仅供学习所用，请于下载后24小时内删除，否则引起的任何后果均由您自己承担。本书版权归原作者所有，如果您喜欢本书，请购买正版支持作者。***

目录

[前言 7](#_Toc285723776)

[主题 7](#_Toc285723777)

[目标读者 7](#_Toc285723778)

[Linux和UNIX 8](#_Toc285723779)

[使用和组织 8](#_Toc285723780)

[例子程序 9](#_Toc285723781)

[练习 10](#_Toc285723782)

[标准和可移植性 10](#_Toc285723783)

[Linux内核和C库版本 11](#_Toc285723784)

[其它语言使用编程接口 11](#_Toc285723785)

[关于作者 11](#_Toc285723786)

[致谢 11](#_Toc285723787)

[许可 12](#_Toc285723788)

[网站和例子程序源代码 12](#_Toc285723789)

[反馈 12](#_Toc285723790)

[第1章 历史和标准 13](#_Toc285723791)

[1.1 UNIX和C简史 13](#_Toc285723792)

[第2章 基础概念 16](#_Toc285723793)

[第3章 系统编程概念 17](#_Toc285723794)

[第4章 文件I/O：统一的I/O模型 18](#_Toc285723795)

[第5章 文件I/O：更多细节 19](#_Toc285723796)

[第6章 进程 20](#_Toc285723797)

[第7章 内存分配 21](#_Toc285723798)

[第8章 用户和组 22](#_Toc285723799)

[第9章 进程凭证 23](#_Toc285723800)

[第10章 时间 24](#_Toc285723801)

[第11章 系统限制和选项 25](#_Toc285723802)

[第12章 系统和进程信息 26](#_Toc285723803)

[第13章 文件I/O缓冲 27](#_Toc285723804)

[第14章 文件系统 28](#_Toc285723805)

[第15章 文件属性 29](#_Toc285723806)

[第16章 扩展属性 30](#_Toc285723807)

[第17章 访问控制列表 31](#_Toc285723808)

[第18章 目录和链接 32](#_Toc285723809)

[第19章 监控文件事件 33](#_Toc285723810)

[第20章 信号：基础概念 34](#_Toc285723811)

[第21章 信号：信号处理器 35](#_Toc285723812)

[第22章 信号：高级特性 36](#_Toc285723813)

[第23章 定时器和睡眠 37](#_Toc285723814)

[第24章 进程创建 38](#_Toc285723815)

[第25章 进程结束 39](#_Toc285723816)

[第26章 监控子进程 40](#_Toc285723817)

[第27章 程序执行 41](#_Toc285723818)

[第28章 进程创建和程序执行的更多细节 42](#_Toc285723819)

[第29章 线程：介绍 43](#_Toc285723820)

[第30章 线程：同步 44](#_Toc285723821)

[第31章 线程：线程安全和线程存储 45](#_Toc285723822)

[第32章 线程：线程取消 46](#_Toc285723823)

[第33章 线程：更多细节 47](#_Toc285723824)

[第34章 进程组、会话和任务控制 48](#_Toc285723825)

[第35章 进程优先级和调度 49](#_Toc285723826)

[第36章 进程资源 50](#_Toc285723827)

[第37章 Daemon 51](#_Toc285723828)

[第38章 编写安全的特权程序 52](#_Toc285723829)

[第39章 能力 53](#_Toc285723830)

[第40章 登录会计 54](#_Toc285723831)

[第41章 共享库基础 55](#_Toc285723832)

[第42章 共享库高级特性 56](#_Toc285723833)

[第43章 进程间通信简介 57](#_Toc285723834)

[第44章 管道和FIFO 58](#_Toc285723835)

[第45章 System V IPC介绍 59](#_Toc285723836)

[第46章 System V消息队列 60](#_Toc285723837)

[第47章 System V信号量 61](#_Toc285723838)

[第48章 System V共享内存 62](#_Toc285723839)

[第49章 内存映射 63](#_Toc285723840)

[第50章 虚拟内存操作 64](#_Toc285723841)

[第51章 POSIX IPC介绍 65](#_Toc285723842)

[第52章 POSIX消息队列 66](#_Toc285723843)

[第53章 POSIX信号量 67](#_Toc285723844)

[第54章 POSIX共享内存 68](#_Toc285723845)

[第55章 文件锁 69](#_Toc285723846)

[第56章 Sockets：介绍 70](#_Toc285723847)

[第57章 Sockets：UNIX Domain 71](#_Toc285723848)

[第58章 Sockets：TCP/IP网络基础 72](#_Toc285723849)

[第59章 Sockets：Internet Domain 73](#_Toc285723850)

[第60章 Sockets：服务器设计 74](#_Toc285723851)

[第61章 Sockets：高级主题 75](#_Toc285723852)

[第62章 终端 76](#_Toc285723853)

[第63章 可选I/O模型 77](#_Toc285723854)

[第64章 伪终端 78](#_Toc285723855)

[附录A：跟踪系统调用 79](#_Toc285723856)

[附录B：解析命令行参数 80](#_Toc285723857)

[附录C：转换NULL指针 81](#_Toc285723858)

[附录D：内核配置 82](#_Toc285723859)

[附录E：更多信息来源 83](#_Toc285723860)

[附录F：部分习题解答 84](#_Toc285723861)

[参考书目 85](#_Toc285723862)

[索引 86](#_Toc285723863)

# 前言

## 主题

本书描述Linux编程接口——Linux（UNIX操作系统的一种免费实现）提供的系统调用、库函数、和其它底层接口。这些接口被直接或间接地使用在Linux上运行的每个程序中。它们允许应用程序完成各种任务：如文件I/O、创建删除文件和目录、创建新进程、执行程序、设置定时器、本机进程和线程间通信、通过网络连接的不同机器进程间通信等等。这些底层接口有时候也叫做系统编程接口。

尽管本书关注于Linux，但我也非常注意标准和可移植性问题，清晰地区分了Linux特有的接口、多数UNIX实现共有的特性、以及POSIX和Single UNIX Specification标准定义的特性。因此本书也提供了UNIX/POSIX编程接口的详尽描述，能够适用于编写UNIX系统应用或跨平台应用的程序员。

## 目标读者

本书主要面向以下读者：

* 为Linux、UNIX、或者其它遵循POSIX的系统开发应用的程序员和软件设计师；
* 在Linux、UNIX、或其它操作系统之间移植应用的程序员；
* Linux或UNIX系统编程课程的教师和高年级学生；
* 希望深入理解Linux/UNIX编程接口，以及系统软件是如何实现的系统管理员和“高级用户”。

我假设你拥有一定的编程经验，但不要求系统编程经验。我还假设你了解C编程语言，并且知道如何使用shell和常用的Linux或UNIX命令。如果你是Linux/UNIX的新手，你会发现第2章非常有用，我们以程序员的视角来讲述Linux和UNIX的基础概念。

## Linux和UNIX

本书原本可以纯粹地讲解标准UNIX（也就是POSIX）系统编程，因为UNIX和Linux的大多数特性都是相同的。不过虽然编写可移植程序是很好的目标，理解Linux对标准UNIX编程接口的扩展也是非常重要的。理由之一是Linux非常流行；其二是有时候为了性能、或使用标准UNIX没有的功能，我们不得不使用非标准的扩展（所有UNIX实现都提供类似的非标准扩展）。

因此本书在适用于标准UNIX的程序员时，还提供了Linux特定编程特性的详细描述。这些特性包括：

* epoll，获得文件I/O事件通知的机制；
* inotify，监控文件和目录改变的机制；
* 能力，授予进程一组超级用户能力的机制；
* 扩展属性；
* i-node标志；
* clone()系统调用；
* /proc文件系统
* Linux对文件I/O、信号、定时器、线程、共享库、进程间通信、和socket的特殊实现细节。

## 使用和组织

你至少可以按两种方式来使用本书：

* 作为Linux/UNIX编程接口的介绍手册。你可以从头到尾阅读本书。后续章节建立在之前章节的基础之上，我尽量避免依赖后续章节的情况。
* 作为Linux/UNIX编程接口的索引参考手册。详细的索引和频繁的交叉引用，允许你随机地阅读任何主题。

我把本书分为以下几部分：

1. 背景和概念：UNIX、C和Linux的历史；UNIX标准简介（第1章）；以程序员的视角介绍Linux和UNIX的基本概念（第2章）；Linux和UNIX系统编程的基本概念（第3章）。
2. 系统编程接口的基础特性：文件I/O（第4章和第5章）；进程（第6章）；内存分配（第7章）；用户和组（第8章）；进程凭证（第9章）；定时器（第10章）；系统限制和选项（第11章）；获取系统和进程信息（第12章）。
3. 系统编程接口的高级特性：文件I/O缓冲（第13章）；文件系统（第14章）；文件属性（第15章）；扩展属性（第16章）；访问控制列表（第17章）；目录和链接（第18章）；监控文件事件（第19章）；信号（第20章到第22章）；定时器（第23章）。
4. 进程、程序、和线程：进程创建、进程结束、监控子进程、执行程序（第24章到第28章）；POSIX线程（第29章到第33章）。
5. 进程和程序的高级主题：进程组、会话、任务控制（第34章）；进程优先级和调度（第35章）；进程资源（第36章）；daemon（第37章）；编写安全的特权程序（第38章）；能力（第39章）；登录会计（第40章）；共享库（第41章到第42章）。
6. 进程间通信（IPC）：IPC简介（第43章）；管道和FIFO（第44章）；System V IPC——消息队列、信号量、共享内存（第45章到第48章）；内存映射（第49章）；虚拟内存操作（第50章）；POSIX IPC——消息队列、信号量、共享内存（第51章到第54章）；文件锁（第55章）。
7. Socket和网络编程：IPC和socket网络编程（第56章到第61章）。
8. 高级I/O主题：终端（第62章）；可选I/O模型（第63章）；伪终端（第64章）。

## 例子程序

我用短小但完整的例子程序来阐述多数接口的使用方法，这些例子都被设计为很容易就能从命令行体验，来查看不同的系统调用和库函数如何工作。所以本书包含大量的示例代码——大概15000行C代码和shell会话日志。

尽管阅读和试验例子程序是不错的起点，掌握本书讨论的概念最有效的方法是编写代码，按你的想法修改例子程序，或者编写新程序都可以。

本书的所有源代码都可以在网站上下载。源代码包含许多书中没有的程序。这些程序的目的和细节在注释中都有相关描述。我提供了Makefile编译这些程序，以及一个README文件，给出了例子程序更多的细节信息。

源代码采用GNU Affero 通用公共授权版本3，可以自由分发和修改。源代码中也包含一份该协议的拷贝。

## 练习

多数章节都以一组练习结束，其中一些是要你按不同方式来试验例子程序，另外一些是该章讨论过的概念相关的问题，还有就是要求你来编写代码以巩固你对本书的理解。你可以在附录F找到部分练习的解答。

## 标准和可移植性

贯穿整本书，我都对可移植性问题特别地关注。你会发现很多相关标准的引用，特别是POSIX.1-2001和Single UNIX规范版本3（SUSv3）标准。同时你还将看到这些标准最新修订的细节改变，也就是POSIX.1-2008和SUSv4标准。（由于SUSv3是更大的修订版本，也是本书编写时最广泛有效的UNIX标准，本书讨论的标准大多是SUSv3，并标注出SUSv4不同的地方。除非我明确地提到，你可以假设我们对SUSv3规范的描述也适用于SUSv4）。

对于那些不是标准的特性，我会指出在不同UNIX实现间的差别。我还会突出那些Linux特定的特性，以及Linux与其它UNIX对系统调用和库函数实现上的细小差别。当某个特性我没有明确指出是Linux专有时，你也通常可以假设它在多数或所有UNIX上都有实现。

本书大多数例子程序我都在Solaris、FreeBSD、Mac OS X、Tru64 UNIX、和HP-UX上测试通过（除了那些Linux独有的特性）。为了提高代码在这些系统上的可移植性，本书网站上提供的某些例子程序有一些额外的代码。

## Linux内核和C库版本

本书主要关注Linux 2.6.x系列，这是本书写作时最广泛使用的内核版本。Linux 2.4的某些细节也会提到，我也会指出Linux 2.4和2.6的区别。当Linux 2.6.x系列出现了新特性时（例如2.6.34），我也会特别指出相应的内核版本号。

至于C库，本书则主要关注于GNU C库（glibc）版本2。当然，glibc 2.x系列版本存在差异时，我也会特别指出。

在本书即将印刷时，Linux内核刚刚发布了2.6.35版本，glibc则已经发布2.12版本。本书完全适用于这两个软件版本。Linux内核和glibc将来接口的变化，会在本书的网站上列出。

## 其它语言使用编程接口

尽管例子程序用C语言编写，你也可以在其它编程语言中使用本书讨论的接口——例如编译型语言C++、Pascal、Modula、Ada、FORTRAN、D；解释型语言Perl、Python、Ruby等。（Java则需要采用一种不同的方式JNI）。不同的语言要获取必要的常量定义和函数声明，需要使用不同的技术（C++除外），另外传递函数参数时可能也需要一点额外的工作。此外就没有太大的区别了，核心概念其实都是一样的。因此即使你使用其它的编程语言，你也会发现本书提供的信息是适用的。

## 关于作者

（略）

## 致谢

（略）

## 许可

电子工程学会和开放组织非常友好地许可我引用IEEE Std 1003.1, 2004版本，以及信息技术标准——可移植操作系统接口（POSIX），开放组织基本规范Issue6。完整的标准可以在<http://www.unix.org/version3/online.html>上在线查阅。

## 网站和例子程序源代码

你可以在<http://www.man7.org/tlpi>上找到关于本书更多的信息，包括勘误表和例子程序的源代码。

## 反馈

我非常欢迎代码bug报告、代码改进建议、以及代码可移植性的提高。同样我也欢迎本书的bug报告和改进建议。由于Linux编程接口总是在变化，我也非常高兴能获得关于本书将来版本的改进意见，包括新特性和变化特性。

Michael Timothy Kerrisk

Munich, Germany and Christchurch, New Zealand

August 2010

[mtk@man7.org](mailto:mtk@man7.org)

# 第1章 历史和标准

Linux是UNIX操作系统家族的成员之一。在计算机的术语里，UNIX已经拥有很悠久的历史。第1章的前半部分简述UNIX的历史。我们首先描述UNIX系统和C编程语言的起源，然后讲述导致Linux发展成为今天这个样子的两个关键因素：GNU项目和Linux内核的开发。

UNIX系统最显著的特点之一是它的开发不是被一个厂商或组织控制。相反许多商业和非商业组织都为UNIX的发展做出了贡献。UNIX也因此增加了许多革新的特性，但同时也导致UNIX各个实现之间的分歧越来越大，编写一个能运行于所有UNIX实现的应用也变得非常困难。于是产生了UNIX的标准化运动，我们将在本章后半部分进行讨论。

## 1.1 UNIX和C简史

第一个UNIX由贝尔实验室（电话公司AT&T的一个部门）的Ken Thompson在1969年开发完成（Linus Torvalds也正是在这一年出生）。这个UNIX是用汇编为Digital PDP-7微计算机编写。UNIX这个名字和MULTICS（Multiplexed Information and Computing Service）有关，后者是AT&T与麻省理工学院（MIT）和通用电子之前合作开发的操作系统项目。（由于该项目最初的失败，没有能够开发出一个有用的系统，当时AT&T已经退出项目）。Thompson的新操作系统从MULTICS中借用了一些设计，包括树型结构文件系统、对命令解释执行采用独立的程序（shell）、以及把文件当作无结构的字节流。

在1970年， UNIX使用汇编语言为新的Digital PDP-11微计算机重新编写，这个PDP-11的遗留痕迹至今仍然可以在多数UNIX实现中找到，包括Linux。

不久之后，Dennis Ritchie，Thompson在贝尔实验室的一个同事，设计和实现了C编程语言。这是一个进化的过程，C起源于更早的解释语言B，最初由Thompson实现了B语言，并从一个更早的语言BCPL中借鉴了许多想法。到1973年，C已经成熟到UNIX内核几乎可以全部使用其重写。UNIX也因此成为最早使用高级语言编写的操作系统，使其迁移到其它硬件体系架构成为可能的重要因素。

C语言的这个起源，解释了C和C++成为今天最广泛的系统编程语言的原因。之前广泛使用的语言都是为其它目的而设计的：FORTRAN为工程师和科学家完成数学任务；COBOL为商业系统处理面向记录的数据流。C填补了一个空白，和FORTRAN、COBOL不一样的是，C语言是几个人为了一个目标而设计的：开发一个高级语言来实现UNIX内核和相关的软件。和UNIX操作系统本身一样，C由专业的程序员为自身所设计。所产生的语言是小巧、高效、强大、简洁、模块化、注重实效、和一致的。

UNIX第一至第六版

在1969年到1979年间，UNIX发布了一系列版本。本质上就是AT&T对UNIX开发进展的一个快照。UNIX最初的六个版本发布时间如下：

* 第一版，1971年11月：此时UNIX还运行在PDP-11上，已经拥有一个FORTRAN编译器，和许多今天依然在使用的工具，包括ar, cat, chmod, chown, cp, dc, ed, find, ln, ls, mail, mkdir, mv, rm, sh, su, who。
* 第二版，1972年6月：UNIX安装在AT&T内部的10台机器上。
* 第三版，1973年2月：这个版本包含一个C编译器和管道的最初实现。
* 第四版，1973年11月：第一个几乎全部用C编写的版本。
* 第五版，1974年6月：此时UNIX已经安装在超过50个系统中。
* 第六版，1975年5月：这是第一个在AT&T范围外广泛使用的版本。

在这些版本发布的过程中，UNIX的使用和声望得到了扩展，首先在AT&T内部，随后在外部。Communications of the ACM杂志发表的一篇关于UNIX的论文也为此做出了巨大贡献。

当时AT&T正在接受美国电话系统对其垄断的政府制裁。AT&T与美国政府的协议禁止其销售软件，这也意味着AT&T不能把UNIX作为产品销售。相反，从1974年的第五版开始，特别是第六版，AT&T授权大学免费使用UNIX。针对大学的UNIX发布版包含文档和内核源代码（当时大约10000行）。

AT&T对大学发布UNIX极大地促进了UNIX的使用和流行，到1977年UNIX已经运行在500个地方，包括125所美国大学和其它一些国家。当时的商业操作系统非常昂贵，而UNIX为大学提供了一个交互式多用户的操作系统，即便宜又强大。同时UNIX还给大学计算机科学研究提供UNIX操作系统的源代码，他们可以修改并提供给学生学习和体验。很多学生学习了UNIX之后，就成为了UNIX的布道者。其它则加入或组建自己的公司，销售运行着UNIX操作系统的计算机工作站。

**BSD和System V的诞生**

1979年1月UNIX发布了第七版，改进了系统的可靠性，提供了一个增强的文件系统。这个发布版还包含一些新的工具，包括：awk, make, sed, tar, uucp, Bourne shell, 和FORTRAN 77编译器。第七版的发布对于UNIX来说具有重要意义，因为从这一刻起，UNIX产生了两个重要的变种：BSD和System V，它们的起源我们马上就会简要地描述。

Thompson在1975/1976学年回到自己的母校，加州大学伯克利分校担任客座教授。在那里他和几个毕业生为UNIX增加了许多新特性。（其中一个学生Bill Joy，随后与别人一起组建了Sun Microsystems，成为UNIX工作站市场早期参与者）。Berkeley开发了许多新的工具和特性，包括C shell、vi编辑器、改进的文件系统（Berkeley Fast File System）、sendmail、Pascal编译器、新的Digital VAX体系架构下的虚拟内存管理等。

在Berkeley Software Distribution（BSD）的授权许可下，这个版本的UNIX，包括它的源代码，被广泛地发布出去。1979年发布了第一个完整发行版3BSD（更早的Berkeley-BSD和2BSD，只是增加Berkeley开发的新工具，而不是完整的UNIX发行版）。

到1983，加州大学伯克利的计算机系统研究组织（Computer Systems Research Group）发布了4.2BSD。这是一个重大的发行版，因为它包含了完整的TCP/IP实现，包括socket应用编程接口（API）和许多网络工具。4.2BSD和它的前任4.1BSD被广泛发布于全世界的许多大学。它们也构成了Sun公司的UNIX变种，SunOS（1983首次发布）的基础。其它重要的BSD发布包括1986年的4.3BSD，以及1993年的最终发布版：4.4BSD。

与此同时，US反托拉斯诉讼强制AT&T解散（法律诉讼起于1970年代中期，1982年解散生效），由于在电话系统中不再垄断，公司被允许运营UNIX。结果就是1981年System III的诞生。AT&T的UNIX支持组（USG）负责开发System III，它雇佣了数百名开发者来增强UNIX，和开发UNIX应用（著名的有document preparation package和软件开发工具）。随后在1983年发布了System V(5)的第一个版本，一系列的小发布版后最终是1989年的System V发布版4（SVR4），到这时System V已经吸收了BSD的许多特性，包括网络基础设施。System V授权给许多商业厂商，这些厂商使用System V作为自己UNIX实现的基础。

因此

# 第2章 基础概念

# 第3章 系统编程概念

# 第4章 文件I/O：统一的I/O模型

# 第5章 文件I/O：更多细节

# 第6章 进程

# 第7章 内存分配

# 第8章 用户和组

# 第9章 进程凭证

# 第10章 时间

# 第11章 系统限制和选项

# 第12章 系统和进程信息

# 第13章 文件I/O缓冲

# 第14章 文件系统

# 第15章 文件属性

# 第16章 扩展属性

# 第17章 访问控制列表

# 第18章 目录和链接

# 第19章 监控文件事件

# 第20章 信号：基础概念

# 第21章 信号：信号处理器

# 第22章 信号：高级特性

# 第23章 定时器和睡眠

# 第24章 进程创建

# 第25章 进程结束

# 第26章 监控子进程

# 第27章 程序执行

# 第28章 进程创建和程序执行的更多细节

# 第29章 线程：介绍

# 第30章 线程：同步

# 第31章 线程：线程安全和线程存储

# 第32章 线程：线程取消

# 第33章 线程：更多细节

# 第34章 进程组、会话和任务控制

# 第35章 进程优先级和调度

# 第36章 进程资源

# 第37章 Daemon

# 第38章 编写安全的特权程序

# 第39章 能力

# 第40章 登录会计

# 第41章 共享库基础

# 第42章 共享库高级特性

# 第43章 进程间通信简介

# 第44章 管道和FIFO

# 第45章 System V IPC介绍

# 第46章 System V消息队列

# 第47章 System V信号量

# 第48章 System V共享内存

# 第49章 内存映射

# 第50章 虚拟内存操作

# 第51章 POSIX IPC介绍

# 第52章 POSIX消息队列

# 第53章 POSIX信号量

# 第54章 POSIX共享内存

# 第55章 文件锁

# 第56章 Sockets：介绍

# 第57章 Sockets：UNIX Domain

# 第58章 Sockets：TCP/IP网络基础

# 第59章 Sockets：Internet Domain

# 第60章 Sockets：服务器设计

# 第61章 Sockets：高级主题

# 第62章 终端

# 第63章 可选I/O模型

# 第64章 伪终端

# 附录A：跟踪系统调用

# 附录B：解析命令行参数

# 附录C：转换NULL指针

# 附录D：内核配置

# 附录E：更多信息来源

# 附录F：部分习题解答

# 参考书目

# 索引