SAMS

GNU/Linux 编程指南

（第二版）

入门·应用·精通

［美］Kurt Wall　等著
张　辉　　译

清华大学出版社

(京)新登字 158 号

著作权合同登记号：01-2001-2327

内 容 提 要

本书全面而深入地介绍了 GNU/Linux 编程。首先介绍了在 Linux 上编程必备的编程工具，然后在库函数、系统调用以及内核上阐述 Linux 编程知识，并专门讲述了包括 TCP/IP、UDP 以及多播套接口在内的网络编程知识；图形界面也是本书的重点内容，本书着重讲述了文本形式的图形界面库 ncurses；还分别讲解了真正图形化的流行系统 X Window、Qt、GNOME 以及 OpenGL 的基本编程方法；最后，介绍了 Bash 编程和设备驱动编程。

本书包含大量实用实例，读者可以通过实例代码深入理解编程思想和技巧。本书另一优点是讲述了其他编程书籍通常没有提及的 RPM 包管理工具、文档编写以及发布许可证选择等内容，这是任何准备投身于 GNU 开发工作的程序员所必须具备的知识。

本书对于所有 Linux 编程人员——无论是初学者还是高级用户——都是一本不可多得的参考资料。

书　　名：GNU/Linux 编程指南（第二版）
作　　者：Kurt Wall, et al.
译　　者：张辉
出 版 者：清华大学出版社（北京清华大学校内，邮编 100084）
印 刷 者：北京市耀华印刷有限公司
发 行 者：新华书店总店北京科技发行所
开　　本：787×1092 1/16　印张：42.125　字数：1024 千字
版　　次：2002 年 6 月第 1 版　　2002 年 6 月第 1 次印刷
书　　号：ISBN 7-302-05550-5/TP・3273
印　　数：0001～5000
定　　价：68.00 元

引 言

很难夸大 Linux 所取得的不可思议的成功和流行。这种成功引发了对开发 Linux 应用的程序员的巨大需求，而且从长远来看，这一需求只会增加。但是，知道从哪里开始，如何开始学习为 Linux 编写程序则很困难。要对新的 Linux 程序员说的是，欢迎加入这场革命！读者将会发现本书在大多数方面是 Linux 编程的优秀指导和教材。如果读者是一位有经验的 Linux 程序员，本书也同样适用，因为它传达了一系列 Linux 编程的主题，其中也包括读者可能尚未探索过的主题。

本书的目的

本书打算向读者展示如何用 Linux 编程，如何在 Linux 上编程以及如何为 Linux 编程。它把注意力几乎全部放在了 C 语言上，因为 C 语言仍是 Linux 的“万国通用语言”。在向读者介绍一些基本开发工具之后，本书立即投入到使用 Linux I/O 模型进行编程的内容当中。本书的第 3 部分介绍了进程和同步的问题，包括线程、内存管理以及进程间通信。本书的第 5 部分专门讨论 Linux 的用户界面，它既用到了基于文本的工具，也用到了基于图形的工具（X Window 系统）。其他部分则涉及到包括 shell 编程和编写设备驱动程序在内的各种各样的话题。本书结尾的 3 章正常情况下会被一般编程的书籍所忽略：它向用户交付应用。这几章向读者展示了如何使用诸如 RPM 这样的包管理工具，以及如何创建有用的文档，还介绍了许可证的问题及其选择。读完本书，读者们就能为投身于称为“Linux”的伟大的社会性以及技术性现象做好了准备。

本书的读者

熟悉其他操作系统但是刚刚接触 Linux 的程序员会得到有关 Linux 编程的详细介绍。读者会接触到将要用到的工具以及将要开展工作的环境。

有经验的 UNIX 程序员会发现 Linux 的编程习惯用法非常熟悉。对于这类读者，本书突出了 Linux 和 UNIX 专有版本的不同之处。最大的可移植性将会是一个重要的话题，因为 Linux 运行在一种尚在变化的平台上：Intel i386，Sun Sparc 和 UltraSparc，Digital Alpha，MIPS 处理器，Power PC 以及基于 Motorola 68000 的 Macintosh 计算机。

中等水平的 C 程序员也能从本书获益。总体而言，在 Linux 编程类似于在其他类 UNIX 系统上编程，所以读者将会很快入门，成为高效的 UNIX 程序员并且理解 Linux/UNIX 深入的特质。

目 录

第 1 部分 Linux 编程工具包

第 2 部分 输入、输出、文件和目录

第 3 部分 进程和同步

第 4 部分 网络编程

第 5 部分 用户界面编程

第 6 部分 特殊编程技术

第 7 部分 补充内容

第 1 章　Linux 及 Linux 编程综述

Linux 不再是爱好者的玩具了。它已经成为几乎每一种计算基础设施，尤其是因特网必不可少的组成部分。如果说编写本书第一版的 1998 年是 Linux 最终出现在美国的雷达屏幕上的一年，那么到了 1999 年 Linux 已经是一名美国正式公民了。再向前看，在 2000 年里，Linux 肯定会把自己稳固树立为因特网计算基础设施的一个基本组成部分以及新经济的一名正式成员。

1.1　Linux 变得成熟了

熟悉 Linux 的程序员尤其是应用程序开发人员都知道 Linux 市场正在爆炸性地增长。本节简要说明了这一情况发生的原因。如果你熟悉 Linux 的历史，可以直接跳到下一节。

1.1.1　Linux 的昨天

Linux 的发布历史始于 1991 年 8 月发表在 Usenet 新闻组 comp.os.minix 上的如下一篇文章，作者是一名芬兰的大学生：

```
Hello everybody out there using minix-
I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones. This has been brewing
since april, and is starting to get ready. I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).
```

这个学生就是 Linus Torvalds，而他所编写的“业余爱好”Linux 已经发展成为现在的 Linux。Linux 1.0 版内核发布于 1994 年 3 月 14 日。1996 年 6 月发布了 2.0 版，在 1.0 版和 2.0 版内核之间，大量的开发工作极大地提高了基本的内核功能、增加了设备支持，最重要的是扩大了可以使用的应用程序的范围。到了发布 2.0 版内核的时候，几乎没有用 Linux 执行不了的任务，只是界面尚不理想（理想的意思是“类似于 Windows”）。

注意： 实际上，Linus 最初关于 Linux 的文章出现在 1991 年 7 月 3 日，但当时并没有特别提到 Linux。Linus 本人讲述的 Linux 早期开发的一段有趣的历史可以从网上得到：http://www.li.org/li/linuxhistory.shtml。

目前稳定的2.2版内核正式发布于1999年1月25日。编写本书的时候，2.3版的开发版内核处于代码冻结状态。此时内核不再增加新特性，按照发布2.4版内核的目标，代码编写工作转向排错并稳定内核的基础代码。该版本的目标是包括一个日志型文件系统、使对称多处理机子系统有更好的粒度，以及对POSIX标准更完备的支持。上面的几点介绍没有涵盖增强和排错工作的所有方面。

1998年3月，当Netscape承诺在GNU计划的GPL（General Public License，通用公共许可证）的一个修订版本的基础上公开Netscape Communicator Internet套件的源代码时，Linux一下子进入了大众的视野。同年7月，世界上最大的两家关系数据库厂商Informix和Oracle宣布把他们的数据库产品移植到了Linux上。1998年8月，Intel和Netscape公司购买了Linux发布商中的领头羊Red Hat公司的少量股票，专项资金投入了Linux世界。同时，IBM开始了它的旗舰数据库产品DB/2在Linux上移植版本的beta测试版。这一年中还有许多类似的开发工作在继续实施。

1999年Linux成为主流IT市场中的年青一分子。主要的软硬件厂商继续向Linux公司进行投资，随后很快又投入了风险资金。不出所料，随着Red Hat和VA Linux首先成为上市的Linux公司，从大型投资商注入的资金取得了成功的股票销售行情。

2000年的头几个月，IBM宣布将在它的服务器、台式机以及便携机产品线上预装Linux。实际上，截止到2000年4月，大多数主要的硬件厂商都把Linux作为其所售计算机系统上的一种候选软件。公众对于Linux的兴趣与日俱增，计算机业内几乎每周都有与Linux相关的消息发布。更令人兴奋的是大众媒体对于Linux的关注。实际上，Linux已经不再仅仅是爱好者们的一个玩具了。

与此同时，那个位于雷蒙德[①]市的小[②]软件公司微软再也坐不住了。在微软著名（也可以说臭名昭著）的万圣节文件泄漏之后，迫使该公司对这种迅速发迹的操作系统予以回应。万圣节文档是微软内部的备忘录，该文档详细地分析了微软对于Linux对其市场霸权尤其是服务器操作系统Windows NT的威胁，并且讨论了用以对付Linux挑战的策略。根据万圣节文档制订的策略，微软发起了一个“恐惧、不可靠、怀疑”（FUD，三个字母分别代表Fear、Uncertainty和Doubt）运动，其后果是很快否认Mindcraft公司有关服务器性能的评测报告，并发表违背常识的带有偏见的白皮书“Linux神话”。Linux社群中的大多数人都嘲笑微软的这些伎俩，而在更大范围的计算机世界里，特别是鉴于微软最初的结论曾经还是开放源代码的开发方式要比传统的软件开发模型更优越这一事实，通常很少有人会严肃看待微软关于Linux或开放源代码软件所发表的声明。然而，微软被迫用FUD的策略来对付Linux的事实表明它对于自己的产品没有信心。

注意： Mindcraft公司最初的研究报告载于http://www.mindcraft.com/whitepapers/nts4rhlinux.html。“Linux神话”一文可通过http://www.microsoft.com/NTServer/nts/news/msnw/LinuxMyths.asp在线阅览。

---

① **译者注：** 雷蒙德(Redmond)是位于美国华盛顿州西雅图东北的一个小城，微软总部所在地。

② **译者注：** 众所周知，微软是世界上最大的软件公司，作者在这里用“小”软件公司来形容它，一方面表示作者对微软的鄙视，另一方面也隐含了微软公司名称中的“小”(Micro)。

1.1.2　Linux 的今天

作为一种服务器级的操作系统，Linux 已经成熟了。提供 Web 服务器的 Linux 系统遍布全球，而且越来越多的商业用户使用 Linux 系统提供文件和打印服务。它既被当作邮件服务器的一种候选平台，也被当作一种强壮而安全的防火墙。业界认为 Linux 在因特网服务器市场将会取得更大份额（绝大多数是原 Windows NT 和 Windows 2000 的市场份额），并且预计 Linux 还会在嵌入式设备市场、因特网设备市场和专门性服务器市场占据主导地位。

Linux 的企业级特性，比如支持多处理器、支持大型文件系统、日志型文件系统以及密集型计算和高可用性集群技术，也会逐步成熟。2.2 版内核最多支持 16 个 CPU，比 2.0 版最多支持 4 个 CPU 有了提高。密集型计算集群技术让 Linux 用户可以创建包含数十乃至数百台廉价的日用型个人计算机的系统，从而达到超级计算机水平的处理速度，而价格同 Cray、SGI 或 SUN 的超级计算机相比极为低廉。高可用性集群技术让 Linux 系统在一个或多个要害部件（比如电源或硬盘）出现故障时，仍然能够继续正常工作。

桌面上的 Linux 也在继续完善。KDE 桌面提供的图形用户界面（GUI，Graphical User Interface）在易用性和可配置方面都能和微软的 Windows 相媲美。但与 Windows 不同的是，KDE 只是运行在健壮而灵活的操作系统之上为改善视觉效果而加入的一层软件。KDE 强大的命令行界面是用户垂手可得的强大工具。Linux 有不少于四种以上办公软件套件：Applixware、StarOffice 以及作为 KDE 计划一部分的 KOffice 都已投入实际使用。Corel 发行了它的 Linux 办公套件 WordPerfect Office 2000，以及它自己的 Linux 发布版本 Corel LINUX OS。在大量 Linux 应用程序和工具软件的基础上，与微软 Office 类似的办公软件的出现使得 Linux 成为 Windows 在桌面系统上的一个竞争对手。

1.1.3　Linux 的明天

Linux 将会何去何从？Linux 因其强壮、稳定、功能强大的特点而在服务器领域极为流行，但它因为在实用性方面持续遭遇挑战而仍然没有在桌面上超越 Windows。在渐露头角的因特网设备，比如防火墙和路由器市场上，Linux 是充满希望的，但还没有在数量上得到完全验证。缩微版本的 Linux 在嵌入式设备，比如电话和机器控制器市场上逐渐流行起来。随着 Linux 厂商，比如 Caldera、Red Hat、VA Linux 以及 Cobalt 作为上市公司逐步成熟，它们必然会推进自己的商业计划和策略，并指导 Linux 的未来。

1.2　为何选择 Linux 编程

人们为什么会用 Linux 编程，又为什么会为 Linux 编程呢？这个问题的答案可能会和从事 Linux 编程的人数一样多。但我认为，大多数答案可以归结为以下几类。

首先，Linux 编程很有意思——这就是我做这件事的原因。其次，Linux 是自由软件。第三，Linux 是开放的。它没有隐藏的接口，也没有未公开的功能或 API（Application Programming Interface，应用程序编程接口）——在了解操作系统的功能方面，没有人或组