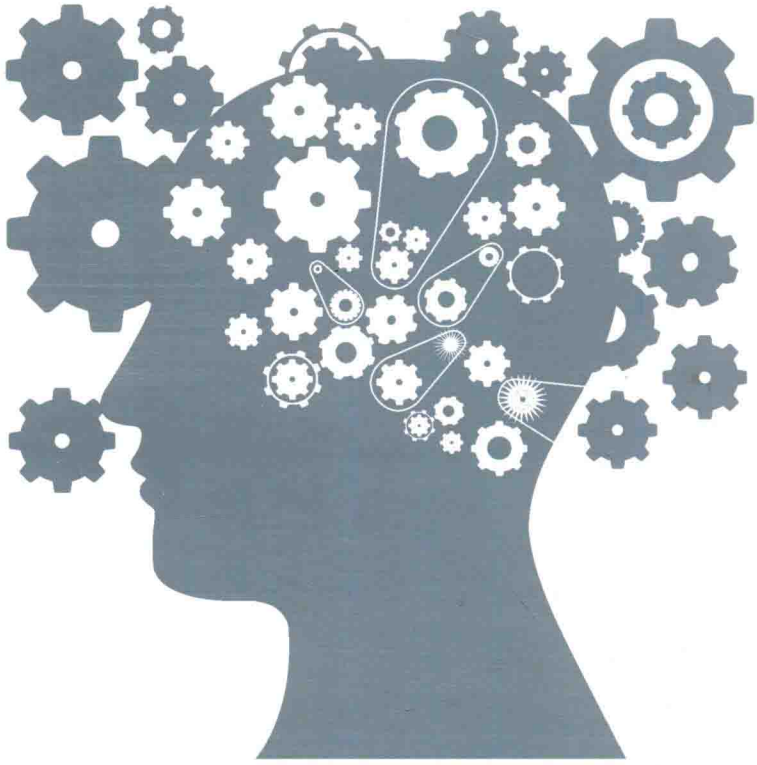
PEARSON

全面深入的 **Linux** 设备驱动程序**名著**

世界级 **Linux** 技术**大师力作**　大量底层**技术内幕**首次公开



Essential Linux
Device Drivers

精通 Linux
设备驱动程序开发

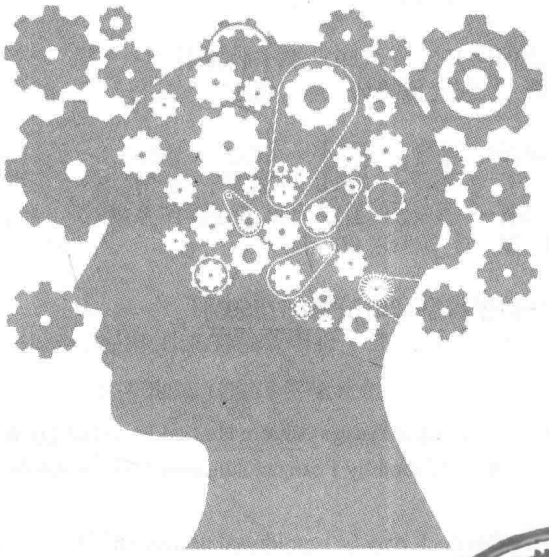
[印] Sreekrishnan Venkateswaran 著

宋宝华 何昭然 史海滨 吴国成 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

PEARSON



Essential Linux
Device Drivers

精通Linux
设备驱动程序开发

[印] Sreekrishnan Venkateswaran 著
宋宝华 何昭然 史海滨 吴国成 译

人民邮电出版社
北京

图书在版编目（CIP）数据

精通Linux设备驱动程序开发 / （印）温卡特斯瓦兰（Venkateswaran, S.）著 ; 宋宝华等译. — 北京 : 人民邮电出版社, 2016.4
ISBN 978-7-115-40251-6

Ⅰ. ①精… Ⅱ. ①温… ②宋… Ⅲ. ①Linux操作系统—驱动程序—程序设计 Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2016)第061559号

版 权 声 明

◆ 著 [印] Sreekrishnan Venkateswaran
译 宋宝华 何昭然 史海滨 吴国成
责任编辑 傅道坤
责任印制 张佳莹 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
北京天宇星印刷厂印刷

◆ 开本：800×1000 1/16
印张：30.25
字数：753 千字 2016 年 4 月第 1 版
印数：1 – 3 000 册 2016 年 4 月北京第 1 次印刷
著作权合同登记号 图字：01-2008-5466 号

定价：89.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

内 容 提 要

本书是Linux设备驱动程序开发领域的权威著作。全书基于2.6内核，不仅透彻讲解了基本概念和技术，更深入探讨了其他书没有涵盖或浅尝辄止的许多重要主题和关键难点，如PCMCIA、I^2C和USB等外部总线以及视频、音频、无线连网和闪存等驱动程序的开发，并讲解了相关的内核源码文件，给出了完整的开发实例。

本书适合中高级 Linux 开发人员阅读。

序

拿到这本书，你也许会问：为什么还要写一本Linux设备驱动程序的书呢？这样的书不是已经有一堆了吗？

答案是：相对于其他同类书，本书实现了一个巨大的突破。

首先，本书与时俱进，基于最新的2.6内核进行讲解。其次，也是更重要的，本书对驱动程序的讲解非常透彻。大多数设备驱动程序的图书仅仅讲解与标准Unix内核或操作系统相关的主题，譬如串口、磁盘驱动和文件系统等，如果你运气好，可能也会碰到讲解网络协议栈的内容。

本书前进了一大步，它没有避重就轻，而是知难而上，探讨了在现代PC和嵌入式系统中必须面对的难点，比如PCMCIA、USB、I^2C、视频、音频、闪存、无线通信等。你可以这样定位本书：Linux内核包含了什么，本书就会告诉你什么。

它毫无遗漏，应有尽有。

何况，作者早就取得了不凡的成就：仅仅是看到他在20世纪90年代末将Linux移植到了手表中的相关介绍就让人激动不已。

本书能成为Prentice Hall开源软件开发系列丛书中的一本，我感到非常激动和欣慰。开源领域从来不乏振奋人心的事件，但本书的面世无疑更加引人瞩目。我希望你能从本书中找到你在进行内核开发时需要的东西，并且也能享受这一过程。

Arnold Robbins

资深Linux技术专家，gawk维护者，Prentice Hall开源软件开发系列丛书编者

前　言

20世纪90年代末，我们IBM的一群同事将Linux内核移植到了一种智能手表上。目标设备虽然微不足道，但是移植Linux的任务却相当艰巨。在当时，内核中还不存在MTD（Memory Technology Device，内存技术设备）子系统，这意味着为了让文件系统能够运行在这种手表的闪存中，我们不得不从头开发必要的存储驱动程序。由于当时内核的输入事件驱动程序接口尚未诞生，因此手表的触摸屏与用户应用程序的接口非常复杂。让X Windows运行在手表的LCD上十分困难，因为X Windows和帧缓冲设备驱动程序搭配得并不好。如果你戴着一块防水的Linux智能手表，却不能躺在浴缸里实时获得股票行情，那么这块手表还有什么用呢？Linux几年前就已集成了蓝牙技术，而当时我们却花费了数月的时间将一种专有的蓝牙协议栈移植到手表上，从而使得这种手表可以联上因特网。电源管理系统虽然只能从手表的电池中多“榨出”短短几个小时时间，但也算够意思了；实际上，为了解决这个棘手的问题，我们也没少花心思。那时候，Linux红外项目Linux-Infrared还不稳定，而为了使用红外键盘输入数据，我们不得不与其协议栈小心翼翼地周旋。最后，由于当时还没有能应用于消费类电子产品的成型的编译器发行版，我们也只能自己编个编译器，并交叉编译出一个紧凑的应用程序集。

时光飞逝，当年的小企鹅已经成长为一名健壮的少年。过去我们编写了成千上万行代码并耗时一年完成的任务，若采用现在的内核，只需要几天就可以完成。但是，要成为一名能巧妙地解决多种问题的高级内核工程师，就必须理解今天的Linux内核提供的各种功能和设施。

关于本书

在Linux内核源代码树提供的各个子系统中，drivers/目录是其中最大的一个分支，它比其他子系统大数倍。随着各种新技术的广泛应用，内核中新的设备驱动程序的开发工作正在稳步加速。最新的Linux内核支持多达70余种设备驱动程序。

本书主要讲解如何编写Linux设备驱动程序，介绍了目前内核所支持的主要设备类型的设计与开发，其中包括当年我在将Linux移植到手表中时未遇到的设备。本书在讲解每类设备驱动程序的时候，都会先介绍与该驱动程序相关的技术，接着给出一个实际的开发例子，最后列出相关的内核源代码文件。在介绍Linux设备驱动程序之前，本书先介绍了内核以及Linux 2.6的重要特性，重点介绍了设备驱动程序编写者感兴趣的内核知识。

读者对象

本书面向渴望在Linux内核上开发新设备驱动程序的中级程序员。要阅读本书，读者需要具备与操作系统相关的基本概念。比如，要知道什么是系统调用，理解为什么在内核开发中需要关注并发问题。本书假定读者已经下载了Linux，浏览过Linux内核源代码，并至少浏览过一些相关的文档。另外，读者必须能非常熟练地使用C语言。

各章概述

前4章为阅读本书其他部分打下了基础，接下来的16章讨论了不同类型的Linux设备驱动程序，之后的第21章描述了设备驱动程序的调试技术，第22章讲解了维护和交付设备驱动程序的相关事宜，最后一章给出了当接到一个新设备驱动程序开发任务的时候，要首先查验的项目清单。

第1章是引言，简单介绍了Linux系统，讲解了下载内核源代码、进行小的代码修改以及建立可启动的Linux内核映像。

第2章引导读者轻松地进入Linux内核的内部结构，讲解了一些必要的内核概念。首先讲述了内核的启动进程，接下来描述了与驱动程序开发相关的内核API，譬如内核定时器、并发管理以及内存分配等。

第3章讲解了对驱动程序开发有用的一系列内核API。首先介绍了内核线程（它提供了一种在内核空间运行后台任务的能力），接下来讲解了一系列的辅助API（如链表、工作队列、完成函数、通知链等）。这些辅助API能简化代码，剔除内核中的冗余，有助于内核的长期维护。

第4章为掌握Linux设备驱动程序开发艺术打基础。这一章首先呈现一般的PC兼容系统和嵌入式设备的体系结构的鸟瞰图，介绍了设备和驱动程序，并讲解了中断处理和内核设备模型等基本的驱动程序概念。

第5章介绍了Linux字符设备驱动程序的体系架构，引入了几个新概念，譬如轮询、异步通知和I/O控制等。由于本书后面介绍的大多数设备都可以看作“超级”字符设备，所以这些概念也与后续章节密切相关。

第6章讲解了内核串行设备驱动程序的层次结构。

第7章讨论了内核中为键盘、鼠标和触摸屏控制器等输入设备服务的输入子系统。

第8章讲解了通过I^2C总线或SMBus总线与系统连接的设备（如EEPROM）的驱动程序，同时也介绍了SPI总线和1-wire总线等其他串行接口。

第9章分析了PCMCIA子系统，讲授了如何编写含PCMCIA或CF组件的设备的驱动程序。

第10章描述了内核对PCI及其衍生总线设备的支持。

第11章探讨了USB的体系架构，并讲解了如何利用Linux内核USB子系统的API来开发USB设备驱动程序。

第12章讲解了Linux视频子系统，分析了内核提供的帧缓冲结构的优点，并给出了帧缓冲设备驱动程序的编写方法。

第13章描述了Linux音频子系统的架构，并给出了音频设备驱动程序的实现方法。

第14章集中描述存储设备（如硬盘）的驱动程序，并介绍Linux块子系统所支持的几种不同的I/O调度策略。

第15章分析了网络设备驱动程序，介绍内核中与网络相关的数据结构以及网络设备驱动程序与协议层接口的实现方法。

第16章描述了各种无线网络设备的驱动程序，如蓝牙、红外、无线局域网WiFi和蜂窝通信等。

第17章讲解了如何让闪存在嵌入式设备上运行起来，这一章最后讲解了PC上的FWH（FirmWare Hub，固件集线器）的驱动程序。

第18章介绍了嵌入式Linux，包括嵌入式设备中的引导装入程序、内核以及设备驱动程序等主要的固件组成。由于Linux在嵌入式领域越来越受欢迎，本书中介绍的Linux驱动程序开发技能极有可能应用于嵌入式领域。

第19章讲解了如何在用户空间驱动各种设备。一些设备驱动程序（尤其是那些重策略、轻性能的设备）更适合在用户空间被驱动。这一章也分析了Linux进程调度对用户空间设备驱动程序响应时间的影响。

第20章描述了之前尚未论及的设备驱动程序系统，如错误侦测和校验（EDAC）、火线接口以及ACPI等。

第21章讲解了用来调试Linux内核代码的各种调试工具，包括跟踪工具、内核探测器、崩溃转储和性能剖析器的使用方法。在开发Linux驱动程序的时候就可用到本章所学的驱动调试技能。

第22章给出了设备驱动程序软件开发生命周期的概况。

第23章给出了当开始进行一个新设备驱动程序开发工作时，应该查验的工作项目清单。本书最后是对“下一步该做什么”的思考。

设备驱动程序中有时候需要以汇编语言实现一些代码片段，因此，附录A介绍了Linux汇编编程的一些内容。x86系统上的一些设备驱动程序直接或间接地依赖于BIOS，因此，附录B讲解了Linux如何与BIOS交互。附录C描述了2.6内核提供的seq文件，它是用于监控和追踪数据点的辅助接口。

本书大体上根据设备和总线的复杂度进行组织，同时也结合了章与章之间互相关联的客观情况。我们从讲解基本的设备类型开始（如字符设备、串口和输入设备），紧接着介绍简单的串行总线（如I^2C和SMBus），之后介绍PCMCIA、PCI和USB等外部I/O总线。由于视频、音频、块和网络设备通常通过这些I/O总线与处理器连接，因此在介绍完这些总线之后，还讲解了这些设备的驱动程序。之后面向嵌入式Linux，讲述了无线连网和闪存等技术。最后讨论了用户空间的设备驱动程序。

内核版本

本书总体上紧跟2.6.23/2.6.24内核版本，书中列出的大部分代码都在2.6.23上测试过。如果读者使用的是更新的版本，请通过类似lwn.net的Linux网站了解内核自2.6.23/2.6.24后进行了哪些更改。

本书网站

我特意建立了elinuxdd.com网站，提供与本书相关的更新和勘误等信息。

本书约定

源代码、函数名和shell命令使用代码体。shell提示符为bash>。新名词使用楷体表示。

为了实现代码示例，一些章节修改了原始的内核源代码。为标识出这些修改，新添加的代码前添加了+，删除的代码前则添加了-。

为简单起见，本书有时使用了通用的路径名。因此，当遇到arch/your-arch/目录时，应该根据当前的编译情况进行转换。例如，如果你正在为x86体系结构编译内核，它应该转换为arch/x86/。类似地，如果你正在为ARM体系结构编译内核，include/asm-your-arch/就应该转换为include/asm-arm/。本书偶尔在文件名中使用*和X作为通配符。因此，如果书中要求查看include/linux/time*.h文件，读者就应该查看include/linux/下的time.h、timer.h、times.h和timex.h所有这些头文件。同样地，如果书中包含类似/dev/input/eventX或/sys/devices/platform/i8042/serioX/这样的文件名，要知道其中的X是指在当前系统配置情况下内核分配给设备的接口号。

→符号有时候会插入在命令或内核的输出之间，目的是附加注释。

为了紧凑地列出函数原型，本书偶尔使用了一些简单的正则表达式。例如，在10.4节中就用pci_[map|unmap|dma_sync]_single()替代了pci_map_single()、pci_unmap_single()和pci_dma_sync_single()。

有几章提到了用户空间的配置文件。例如，第2章打开了/etc/rc.sysinit文件，第16章引用了/etc/bluetooth/pin文件。这些文件的确切名称和位置都有可能因Linux发行版本的不同而有所变化。

致　谢

首先感谢Prentice Hall出版社负责本书的编辑们：Debra Williams Cauley、Anne Goebel和Keith Cline。没有她们的支持，本书就不可能完成。感谢Mark Taub，他最早对本书的主题产生兴趣并策划了这本书。

过去的10年里，很多人和事对我的研究帮助巨大：在各个Linux项目里一起共事的同事们、强健的内核源代码、邮件列表和互联网。这些都对我完成本书起了重大作用。

*Linux Magazine*的Martin Streicher邀请我参与该杂志的“超级发烧友”内核栏目，从而把我从一名全职程序员变成了一名兼职作者。我从他身上学习到了许多技术写作技巧，在此表示感谢！

我要特别感谢我的技术审稿人。Vamsi Krishna耐心地读完了初稿的每一章，提出了很多建设性的建议，令本书增色不少。Jim Lieb对书中的几章提供了有价值的反馈意见。Arnold Robbins浏览了开始的几章并且提供了富有见地的意见。

最后，我要感谢我的父母和妻子，感谢他们的爱与支持。我还要感谢我幼小的乖女儿，正是她不断地提醒我把时间投入到这本书上来：她摇摇晃晃地走来走去，那离奇的步伐极似企鹅。

目　录

第1章 引 言

本章内容

- 演进
- GNU Copyleft
- kernel.org
- 邮件列表和论坛
- Linux发行版
- 查看源代码
- 编译内核
- 可加载的模块
- 整装待发

Linux具有诱人的魅力，它是一个由全世界不同民族、不同信仰、不同性别的人共同参与和协作的国际性项目。Linux免费提供源代码，并且具有与Unix类似的为人们所熟悉的应用程序编程环境，这一切造就了它今天的巨大成功。通过互联网从专家处即时获得的高质量的免费支持也发挥了重要作用，它促成了一个庞大的Linux社区。

在技术方面，开发人员可以获得所有源码，并由此得出一些创新方案，他们因此感到无比振奋。譬如，你可以修改（hack）[①] Linux的源码，并做定制，让设备在几秒钟之内启动，而使用一个有专利的商业操作系统则很难完成这样的壮举。

1.1 演进

1991年，一位名为Linus Torvalds的芬兰大学生开发了Linux操作系统。起初这只是他个人的爱好，但它很快就发展成为在全世界范围内广受欢迎的先进的操作系统。Linux第一次发布时仅支持Intel 386处理器，但是后来，它的内核复杂性逐步增加，可以支持众多的体系架构、多处理器硬件和高性能集群。Linux所支持的体系结构非常多，主要支持的一些硬件架构是x86、IA64、ARM、PowerPC、Alpha、s390、MIPS和SPARC。Linux已经被移植到成千上万的基于这些处理器的硬件平台之上。与此同时，其内核还在不断完善，系统性能也在飞速提升。

① 意指对源码进行一些有针对性的修改。——译者注

虽然开始的时候Linux只是一个桌面操作系统，但目前它已经进入嵌入式和企业级计算领域，并融入了我们的日常生活。当你按动掌上电脑的按键，用遥控器把电视切换到天气频道，或者在医院接受体检的时候，很有可能就在享受某些Linux代码提供的服务。技术优势和开源特性促进了Linux的演进。无论是试图开发不到100美元的计算机以改善世界贫困地区的教育状况，还是要降低消费类电子产品的价格，Linux如今都已成为一个绝好的选择，因为商业操作系统的价格有时候比计算机本身的价格更贵。

1.2　GNU Copyleft

GNU工程比Linux更早诞生，发起它的目标是定制一个免费的类Unix操作系统（GNU是GNU's Not Unix的递归缩写，意为“GNU不是Unix”。一个完整的GNU操作系统基于Linux内核构建，但也包含一些其他组件，如库、编译器和实用程序（utility）。因此，基于Linux的计算机的更准确称呼应该是GNU/Linux系统。GNU/Linux系统的所有组成部分都建立在免费软件之上。

免费软件有许多种，其中的一种是公共领域（public domain）软件。公共领域发布的软件没有版权，对于它的使用也不会强加任何限制。你可以免费使用它，随意修改它，甚至限制别人发布你修改后的代码。发现了吗？所谓“没有限制”条款居然暗含了对下游施加限制的权力。

GNU工程的主要发起者——自由软件基金会——创造了GNU公共许可证（GPL），它也被称为“版权左派”（copyleft）[①]，以防止有人中途将免费软件转化为商业软件。谁修改了copyleft的软件，就必须以copyleft的方式分享他的软件。GNU系统中Linux内核以及大部分组件（如GNU编译器GCC）都以GPL发布。因此，如果你修改了内核，你就必须在社区分享此修改。实际上，你必须以copyleft的形式将授予你的权利传递出去。

> Linux内核基于GPL第2版。在内核社区，人们一直在争论是否应该采用GPL的最新版本GPLv3。目前的趋势似乎是反对采用GPLv3。

通过系统调用访问内核服务的Linux应用程序没有被看作衍生的工作，因此并不受限于GPL。而库则采用GNU轻量级通用公共许可证（LGPL），其限制要少于GPL。商业软件也允许与LGPL下的库动态链接。

1.3　kernel.org

Linux内核源代码主要存放在www.kernel.org。该网站包含所有已经发布的内核版本，世界各地有大量的kernel.org镜像网站。

除了已经发布的内核以外，kernel.org还包含了由一线开发人员提供的补丁，这些补丁可以作为未来稳定版本的试验平台。补丁是一种文本文件，包含了新开发版本和开发之初制订的原始版本之间的源码差异。由Linux内核第一维护人Andrew Morton定期提供的-mm补丁是一种很受欢迎的补丁。在该补丁中，我们可以找到在主线源代码树中尚未提供的实验性的功能。另一个会定期

① 版权为copyright，这里故意用copyleft。但是，copyleft作品是有版权的，只是加入了法律上的分发条款。——译者注