TURING 图灵程序设计丛书 Linux/Unix系列

PRENTICE HALL PTR

The Linux Kernel Primer

Linux内核编程

[美] Claudia Salzberg Rodriguez
Gordon Fischer
Steven Smolski
著

陈莉君 贺炎 刘霞林 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵

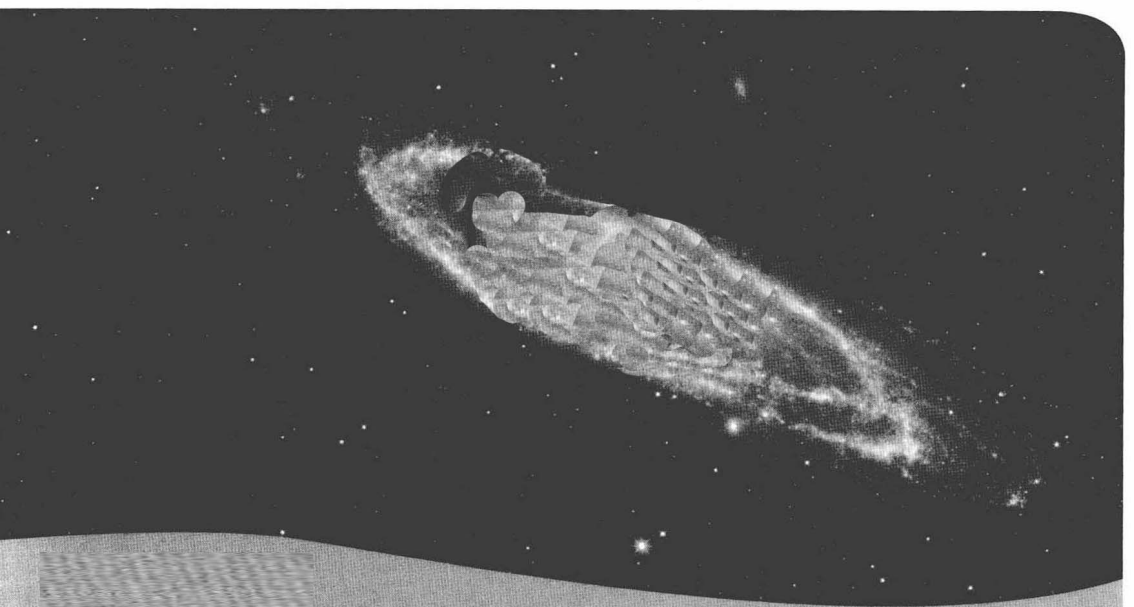
Linux/Unix系列

The Linux Kernel Primer

Linux内核编程

[美] Claudia Salzberg Rodriguez
Gordon Fischer
Steven Smolski
著

陈莉君 贺炎 刘霞林 译



人民邮电出版社
北京

图书在版编目（CIP）数据

Linux内核编程 / (美) 罗德里格斯 (Rodriguez,C.S.) , (美) 费舍尔 (Fischer,G.) , (美) 斯莫斯基 (Smolski,S.) 著 ; 陈莉君, 贺炎, 刘霞林译. -- 北京 : 人民邮电出版社, 2011.6
(图灵程序设计丛书)
书名原文: The Linux Kernel Primer: A Top-Down Approach for x86 and PowerPC Architectures
ISBN 978-7-115-25194-7

Ⅰ. ①L… Ⅱ. ①罗… ②费… ③斯… ④陈… ⑤贺… ⑥刘… Ⅲ. ①Linux操作系统－程序设计 Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2011)第055985号

内 容 提 要

本书以 Linux 操作系统为基础，详细介绍了 Linux 内核子系统，并辅以大量内核源代码和示例程序进行演示，引领读者深入 Linux 内核。本书的主要内容包括：Linux 基础知识、内核探索工具集、进程的整个生命周期、内存区、页面、Slab 分配器、用于输入 / 输出的各种设备、文件系统、抢占、自旋锁、信号量、内核引导、构建 Linux 内核，以及向内核添加代码等，同时还简单介绍了一些应用工具和实用程序。每章末尾都给出一些练习，涉及内核运行的操作及工作原理。

本书适合对 Linux 内核感兴趣的各层次读者，无论对 Linux 初学者还是 Linux 程序开发人员，本书都是一本很有价值的参考书。

图灵程序设计丛书
Linux内核编程

◆ 著 [美] Claudia Salzberg Rodriguez Gordon Fischer Steven Smolski
译 陈莉君 贺 炎 刘霞林
责任编辑 王军花
执行编辑 李 静

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
中国铁道出版社印刷厂印刷

◆ 开本：800×1000 1/16
印张：26
字数：614千字 2011年6月第1版
印数：1－3 000册 2011年6月北京第1次印刷
著作权合同登记号 图字：01-2010-4025号
ISBN 978-7-115-25194-7

定价：75.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版权声明

译 者 序

溯本求源，从本书开始！

打开 Linux 内核源代码，我们可以看到熟悉的 C 语言函数和一些陌生的汇编代码。应该说，Linux 内核入门是不容易的，原因在于其代码规模庞大，而且涉及的层面众多。规模一大，就不易现出本来面目，枝叶庞杂，自然不容易找到着手之处；层面一多，就会让人眼花缭乱，盘根错节，怎能让人提纲挈领？

就我们的经验，本书是内核初学者（不是编程初学者）登堂入室的首选。本书三位作者有多年的行业经验：Claudia Salzberg Rodriguez 就职于 IBM Linux 技术中心，从事内核及相关编程工具的开发工作；Gordon Fischer 为很多设备开发了 Linux 和 UNIX 设备驱动程序；Steven Smolski 在半导体行业已经浸染了 26 年，开发过各种驱动程序和嵌入式系统。他们合作奉献给大家的这本内核入门书，是对 Linux 内核编程的有效指导。作者独特的由表及里的讲解方法使得内核编程更易于理解：从用户空间到内核，把内核内在的实现原理与用户级编程的基本原则相联系，系统地追踪了实现功能。这种途径有助于扩大你所了解的 Linux 知识，加深对内核组成及工作机理的理解。

在本书的翻译过程中，我们能感受到作者软硬件知识之全面、内容组织方式之独到，以及开发经验之丰富。在我们熟知的 x86 平台之外，作者还对 PowerPC 平台进行了深入讲解，不仅让基于 PowerPC 平台的开发者找到了知音，更为 x86 的开发者打开了一扇窗。

本书第 1、4、9、10 章由陈莉君翻译，第 2、5、8 章由贺炎翻译，第 3、6、7 章由刘霞林翻译，全书由陈莉君统稿。书中不妥之处和错误在所难免，望读者指正。

序　言

“有龙出没”，中世纪地图绘制者碰到未知和危险的地方就如此标记，可能你首次敲入如下命令也有这样的感觉：

```
cd /usr/src/linux ; ls
```

“我该从何处入手？”你困惑不已。“我到底在寻找什么？所有这些是怎样放在一起并真正起作用的？”

功能全面的现代操作系统庞大而复杂，有为数不少的子系统，它们之间的交互更是错综复杂而微妙难言。不错，你的确拥有 Linux 内核源代码（稍后还会详述），但是，从何处开始阅读，应留意哪些地方，该以怎样的顺序浏览代码，这些问题的答案都不是显而易见的。

本书的作用正在于此。它一步一步地让你了解到内核的各个部分，它们如何工作，互相之间怎样关联。本书作者熟知内核，将这些知识贯穿于本书的始终。在学完本书之后，你和内核至少会成为好朋友，乃至产生深厚的情意。

Linux 内核是“自由”软件。Richard Stallman 在“自由软件的定义”一文[①]中，说明了软件自由的含义。有两个级别，Freedom 0，运行软件是自由的，这是最基本的自由。紧跟其后的是 Freedom 1，探究程序运行原理也是自由的。这种自由往往被忽略，实际上，这才是最重要的，因为学习的最好方式就是观察别人如何做事。在软件世界中，那就意味着阅读别人的程序，并了解他们哪方面做得好，哪方面做得不好。至少在我看来，GNU/Linux 之所以在现代计算领域变成一股强大的力量，最根本的原因之一就是 GPL 协议的自由。你在使用 GNU/Linux 的每时每刻都会感到这种自由的可贵，别忘了经常默念一下它的好处。

本书充分利用 Freedom 1，带你深入研究 Linux 内核源代码。你会看到所有的一切，有些方面的确做得不错，还有些事也并不尽人意。但是，由于有了 Freedom 1，你会看到全貌，并从中学到很多。

况且，这也使我与“Prentice Hall 开源软件开发系列丛书”结缘，本书是丛书的第一辑。开发这个系列的想法来自于这样一个理念：阅读代码是学习编程的最好方法。如今这个世界，人们幸福地享有丰富而自由的开源软件，这些源代码正期待着（或许热切渴望着）被大家阅读、理解和赞许。该系列丛书旨在成为你软件开发学习过程中的领路人，即通过尽可能多地展示真实的代码帮助你学好编程。

---

① http://www.gnu.org/philosophy/free-sw.html

我真诚地希望你会喜欢本书，并从中获益多多。我也希望本书能激发你的灵感，从而在自由软件和开源世界开创你自己的事业，用这种方式参与进来，那无疑是最令人愉快的了。

祝你学习愉快！

Arnold Robbins
丛书主编

前　言

无论是一般性技术还是计算机技术，对于试图了解它们的人们来说都具有不可思议的魔力。技术的发展使其影响力不断扩大，迫使人们对一些旧的概念重新评估。Linux 操作系统已经对产业变革和商业营销方式转变做出了巨大贡献。它采用 GNU 公共许可证并与 GNU 软件良性互动，占据了中心位置，围绕开源、自由软件和开发社区思想的各种争论都离不开它。Linux 无疑是一个极其成功的典范，展现了开源操作系统无比强大的力量，其理论的魔力令世界各地的程序员们如痴如狂。

对于大多数计算机用户来说，使用 Linux 正变得越来越方便。有了各种各样的发布版、社区的支持，以及工业后盾，Linux 的应用也找到了安全的港湾，出现在大学、行业应用以及数以千计的家庭用户中。

使用大潮促进了技术支持和新功能需求的日益增长。这样一来，愈来愈多的程序员发现自己对 Linux 内核内幕感兴趣，因为大量现有的（还在快速增长的）应用需要支持不同的体系结构和种类繁多的新设备。

内核向 Power 体系结构的成功移植，也助长了 Linux 操作系统在高端服务器和嵌入式系统中的全面繁荣。许多公司购买基于 Power PC 的系统来运行 Linux，因此越来越多的人需要知道 Linux 在该体系结构上的运行机理。

适合的读者

本书的读者包括初级和经验丰富的系统程序员、Linux 的热衷者，以及应用程序的开发者，这些开发者渴望更好地理解自己的程序到底是如何工作的。只要有 C 语言知识，熟悉基本的 Linux 用法，如果想知道 Linux 如何工作，那么你就会发现这本书提供了所需的基本知识，可以说，本书是理解 Linux 内核如何工作的初级读本。

不管你是只登录过 Linux 并编写了些小程序，还是你本身就是一个系统程序员，正想深入理解某个子系统的特性，本书都会有你所要的信息。

内容组织

本书分为三部分，每部分都提供必要的知识，让读者能顺利地钻研 Linux 内幕。

第一部分提供必要的工具和背景，便于对 Linux 内核展开进一步的探索。

第 1 章回顾了 Linux 和 UNIX 的历史，对比了很多发布版，并从用户空间的角度简述各种内

核子系统。

第 2 章描述 Linux 内核中常用的数据结构和语言的用法，介绍 x86 和 PowerPC 体系结构的汇编语言，并简述一些工具和实用程序，可用来获取理解内核内幕所需的信息。

第二部分介绍了在每个内核子系统中所涉及的基本概念，并分析了执行子系统功能的代码。

第 3 章讨论了进程模型的实现。本章解释了为何引入进程，并讨论了进程往返于用户空间和内核空间时的控制流。我们还讨论了进程在内核中是如何实现的，并描述了与进程执行相关的所有数据结构。本章还介绍了中断和异常，描述了这些硬件机制在每种体系结构中是如何发生的，它们与 Linux 内核又是如何交互的。

第 4 章描述了 Linux 内核如何追踪和管理用户空间进程的可用内存和内核的可用内存。本章描述了内核对内存分类的方式，以及决定分配和释放内存的方式，也详细描述了缺页机制以及它怎样在硬件上执行。

第 5 章描述了处理器如何与其他设备进行交互，内核又是如何响应和控制这些交互的。本章还涵盖了各种设备及其在内核中的实现。

第 6 章概述文件和目录如何在内核中实现。本章引入了虚拟文件系统，它是用于支持多文件系统的抽象层。本章还跟踪了文件相关操作的执行，如打开和关闭文件。

第 7 章描述调度程序的操作，调度程序让多个进程运行起来就像系统中只有一个进程在运行一样。本章详细描述了内核如何选择执行哪一个任务，进程切换时如何与硬件进行交互。本章还叙述了什么是内核抢占，它又是怎样执行的。最后，描述了系统时钟的工作原理，内核怎样使用它计时。

第 8 章描述电源开和关时都发生些什么。本章对各种处理器处理内核加载的方式进行了跟踪，包括对 BIOS、Open Firmware 和 bootloader 的描述。然后，考察了内核启动和初始化时的线性顺序，涉及了前面章节中讨论的所有子系统。

第三部分，描述如何编译内核并与内核进行交互的有效途径。

第 9 章涵盖了编译内核所必需的工具链和所执行的对象文件的格式。还详细描述了内核源代码编译（Kernel Source Build）系统如何运作，怎样把配置选项加入内核编译系统中。

第 10 章描述了 `/dev/random` 操作，这在所有的 Linux 系统中都可以看到。本章用它来跟踪设备，并从更具实战性的角度介绍前面各章描述过的概念。最后介绍了如何在内核中实现自己的设备。

我们的探索方法

本书给读者介绍了理解内核的必要概念。我们遵循自顶向下的方式来组织内容，具体体现在以下两个方面。

首先，我们把内核的机理和用户空间操作的执行关联起来，因为读者对后者较熟悉，所以我们会将二者结合起来，解释内核的工作。在可能时，我们从用户空间的例子说起，并跟踪代码的执行到内核。但有时，这种跟踪方式并不有效，因为需要先介绍子系统的数据类型和子结构，而后才能解释其工作原理。在这些情况下，我们把对内核子系统的解释和它与用户空间程序如何联

系的具体例子结合起来。有双重意图：其一，当内核一方面与用户空间打交道，另一方面与硬件打交道时，突出在内核中看到的层面；其二，通过跟踪代码和事件发生的顺序来解释子系统的工作原理。我们相信，这有助于读者将内核的工作原理与自己的认识匹配起来，也有利于读者了解一个特定的功能怎样与操作系统的其余部分产生联系。

其次，我们以自顶向下的角度，考察把数据结构视作子系统操作中心，并观察它们怎样与系统管理的执行行为相联系。我们尽力刻画子系统操作中心的结构，并像追踪子系统的操作一样持续关注这些数据结构。

约定

你会在全书中看到源代码列表。右上角存放有关源代码树根的源文件位置。代码中的行号是为了方便随后对代码进行解释。我们在解释内核子系统及其工作原理时，会不断引用源代码并给予解释。

命令行选项、函数名、函数输出以及变量名都用代码体加以区分。

引入一个新概念时就采用**黑体**。

致　谢

我们在此要感谢很多人，没有他们的帮助，本书不可能出版。

Claudia Salzberg Rodriguez：首先要说的是，有许许多多的人采用了各种各样的方式帮助我们完成了这本书，但面对有限的“致谢”篇幅，却很难将那些做出突出贡献的人一一列举出来。话虽如此，我还是要在此感谢所有对 Linux 内核有所贡献的人，正是他们出于对 Linux 的热爱，用辛勤工作和无私奉献开发出这样好的操作系统。我深深地感谢给予我指点的很多引路人，他们唤醒并培育了我探究一切的好奇心，并教会了我学习的本领。我也要感谢家人永不衰竭的爱与支持，他们的热情帮我度过了疲惫不堪的时期。最后，我衷心感谢 Jose Raul，他总是及时满足我的要求，总能找出办法重新点燃我的灵感。

Gordon Fischer：我要感谢所有的编程人员，在我还是菜鸟时他们能耐心地向我解释错综复杂的 Linux 内核。我还要感谢 Grady 和 Underworld，他们向我提供了美妙的编码音乐。

我们要感谢出色的编辑 Mark L. Taub，他知晓在本书出版过程中的每一步，应该怎样去做，并引领我们在那个方向上不断努力。还要感谢他在本书的撰写过程中，总能做到通情达理、善解人意，既严格要求，又平易近人。

我们还要感谢 Jim Markham 和 Erica Jamison。感谢 Jim Markham 是因为他早期的修改意见为全书的编写奠定了基础。感谢 Erica Jamison 是因为在手稿的最后一版编辑中给我们提供了反馈意见。

我们该感谢审稿人了，他们花大量的时间阅读本书，并给出建设性意见，使本书更加完善。感谢你们敏锐的眼光和深刻的见解，你们的建议是无价的。审稿人（按字母顺序）依次是 Alessio Gaspar、Mel Gorman、Benjamin Herrenschmidt、Ron McCarty、Chet Ramey、Eric Raymond、Arnold Robbins 和 Peter Salus。

我们要感谢 Kayla Dugger，她在整个编辑校对过程中以极大的热情鼓励着我们，还要感谢 Ginny Bess，她有着犀利的编辑眼光。这里要特别感谢参与编辑、校对、排版、市场以及印刷的幕后群体，正是他们，才使本书得以顺利完成。

目　录

第1章 概 述

本章内容

- UNIX的历史
- 标准和通用接口
- 自由软件和开放源码
- Linux发布版概览
- 内核版本信息
- 基于Power的Linux
- 什么是操作系统
- 内核组织
- Linux内核概述
- 可移植性和体系结构的相关性

Linux 操作系统诞生于 1991 年，是 Linus Torvalds 学生时代业余爱好的产物。与当下的迅猛发展势头相比，当初的 Linux 显得微不足道。Linux 刚开发时运行在具有 AT 硬盘、x86 体系结构的微处理器计算机上。第一个发布版支持 bash shell 和 gcc 编译器。那时还不关心其可移植性，也没有设想其在学术界和工业界是否能够广泛应用，更没有商业计划或远景宣言。然而，从第一天起，它就是免费可得的。

从早期的 beta 版开始，Linux 就在 Linus 的指导和维护下，变成了一个协作项目。这填补了一项空白，即黑客们需要一个运行在 x86 体系结构上的免费操作系统。这些黑客们开始贡献源代码，为他们特定的需要提供支持。

通常说，Linux 是一种 UNIX。从技术上说，Linux 是 UNIX 的克隆，因为它实现了 POSIX UNIX 规范 P1003.0。UNIX 从 1969 年诞生以来就统治着非 Intel 工作站领域，被公认为是一个强大而又优雅的操作系统。UNIX 既然定位在高性能工作站，就只在研究机构、学术单位以及开发机构中使用。Linux 把 UNIX 系统的能量带入到了 Intel 个人计算机及其用户家里。如今，Linux 在工业和学术领域得到广泛应用，支持众多的体系结构，如 PowerPC。

本章简要介绍有关 Linux 的概念，引领你概览内核的组成和特性，并介绍一些引人入胜的 Linux 功能。为了理解 Linux 内核的概念，你需要对其开发动机有一个基本理解。

1.1　UNIX的历史

刚刚说过，Linux是一种UNIX。尽管Linux并不是直接从现有的UNIX衍生而来的，但事实上，它实现了通用UNIX标准，因此有必要来看看UNIX的历史。

MULTICS（MULTiplexed Information and Computing Service）被认为是UNIX操作系统的鼻祖，它是麻省理工学院、贝尔实验室和通用电气公司的一家合资企业开发的操作系统，该企业主要从事计算机制造。MULTICS的开发是想让机器支持众多的分时用户。在合资企业成立的1965年，尽管当时操作系统支持**多道程序设计**（可在多个作业之间分时），但依然是只支持单用户的批处理系统。从用户提交一个作业到获得输出之间的响应时间要用小时计算。MULTICS的主要目的是开发一个**多用户分时**系统，让每个用户都可以访问自己的终端。尽管贝尔实验室和通用电气公司最终放弃了这一项目，但MULTICS还是在许多地方得以实际应用。

UNIX的开发始于移植精简的MULTICS版本，从而开发出一个PDP-7小型计算机上的操作系统，让这个新操作系统能支持一种新的文件系统，即UNIX文件系统的第一个版本。由Ken Thompson开发的这个操作系统，支持两个用户，有一个命令解释器和对新文件系统进行文件操作的一组程序。1970年，UNIX被移植到PDP-11上，经修改后能支持更多的用户。这就是第1版的UNIX。

1973年发布的UNIX第4版，由Ken Thompson和Dennis Ritchie用C（由Ritchie刚刚开发的一种语言）重写。这就让操作系统脱离纯汇编语言，并打开操作系统可移植性的大门。想象一下这种转变的意义有多大！当时，操作系统完全是在系统的体系结构规范下建立起来的，因为汇编语言与体系结构密切相关，所以操作系统不易移植到其他体系结构。用C重写UNIX是向更可移植（和可读）的操作系统迈向的第一步，也是让UNIX迅速普及的一步。

1974年是UNIX在大学中迅速普及的头一年。学术机构开始与贝尔实验室的UNIX系统开发组进行合作，开发出具有很多创新特色的第5版。这一版本可免费获得，其源代码可供大学用于教学。1979年，在众多创新、代码简化以及改善可移植性之后，UNIX操作系统第7版（V7）诞生。这一版本包含了C编译器和著名的B shell命令解释器。

20世纪80年代出现了个人计算机。工作站当时只用在企业和大学中。大量UNIX变体从第7版衍生而来。这些变体包括Berkeley UNIX（BSD）和AT&T UNIX System III和System V，其中BSD是由加利福尼亚大学伯克利分校开发的。每个变体又会演变出其他系统，如NetBSD和OpenBSD（BSD的变体）以及AIX（IBM的System V变体）。事实上，UNIX的所有商用变体都来源于System V或BSD。

1991年，Linux出现，那时，UNIX非常流行，但不适用于PC。购买UNIX的价格令人望而却步，的确不是一般用户可以买的，除非他在大学工作。Linux最初是作为Minix操作系统（由Andrew Tanenbaum开发的一个教学用小型操作系统）的扩展而实现的。

随后的几年，Linux内核与FSF（Free Software Foundation，自由软件基金会）的GNU项目所提供的系统软件相结合，使得Linux[①]发展成为非常坚实的系统，吸引着贡献代码的黑客以外的

① 为了体现Linux是FSF的GNU项目所提供的系统软件的组成部分，Linux也被称为GUN/Linux。