

## <<Linux内核编程>>

### 图书基本信息

书名：<<Linux内核编程>>

13位ISBN编号：9787115251947

10位ISBN编号：7115251940

出版时间：2011-5

出版时间：人民邮电

作者：(美)罗德里格斯//费舍尔//斯莫斯基|译者:陈莉君//贺炎//刘霞林

页数：400

译者：陈莉君 贺炎 刘霞林

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Linux内核编程&gt;&gt;

## 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

## &lt;&lt;Linux内核编程&gt;&gt;

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。

## <<Linux内核编程>>

### 内容概要

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。

## <<Linux内核编程>>

### 作者简介

Claudia Salzberg Rodriguez

就职于IBM

Linux技术中心，负责内核及相关编程工具的开发。

在担任程序员的10多年时间里，她一直针对Intel和PPC平台进行Linux嵌入式系统到高性能系统的开发

。

Gordon Fischer

曾为很多底层设备开发过Linux和Unix设备驱动程序，并针对Intel和PPC平台上的企业级系统中使用Linux内核。

Steven Smolski

有近30年的半导体行业从业经验。

从事过内存、处理器以及ASICs的研发和测试，为Linux、AIX以及Windows都编写过应用程序和驱动程序，还进行过嵌入式操作系统的开发。

## &lt;&lt;Linux内核编程&gt;&gt;

## 书籍目录

## 第1章 概述

- 1.1 UNIX 的历史
- 1.2 标准和通用接口
- 1.3 自由软件和开放源码
- 1.4 Linux 发布版概览
  - 1.41 Debian
  - 1.42 Red Hat/Fedora
  - 1.43 Mandriva
  - 1.44 SUSE
  - 1.45 Gentoo
  - 1.46 Yellow Dog
  - 1.47 其他发布版
- 1.5 内核版本信息
- 1.6 基于Power的Linux
- 1.7 什么是操作系统
- 1.8 内核组织
- 1.9 Linux 内核概述
  - 1.9.1 用户接口
  - 1.9.2 用户标识符
  - 1.9.3 文件和文件系统
  - 1.9.4 进程
  - 1.9.5 系统调用
  - 1.9.6 Linux 调度程序
  - 1.9.7 Linux 设备驱动程序
- 1.10 可移植性和体系结构的相关性
- 1.11 小结
- 1.12 习题

## 第2章 内核探索工具集

- 2.1 内核中常见的数据类型
  - 2.1.1 链表
  - 2.1.2 查找
  - 2.1.3 树
- 2.2 汇编
  - 2.2.1 PowerPC
  - 2.2.2 x86
- 2.3 汇编语言示例
  - 2.3.1 x86 中的汇编示例
  - 2.3.2 PowerPC 中的汇编示例
- 2.4 内联汇编
  - 2.4.1 输出操作数
  - 2.4.2 输入操作数
  - 2.4.3 已修改过的寄存器（已修改的元素列表）
  - 2.4.4 参数的编号方式
  - 2.4.5 约束条件
  - 2.4.6 asm

## &lt;&lt;Linux内核编程&gt;&gt;

- 2.4.7 \_\_volatile\_\_
- 2.5 特殊的C 语言用法
  - 2.5.1 asmlinkage
  - 2.5.2 UL
  - 2.5.3 内联
  - 2.5.4 const 和volatile
- 2.6 内核探索工具一览
  - 2.6.1 objdump/readelf
  - 2.6.2 hexdump
  - 2.6.3 nm
  - 2.6.4 objcopy
  - 2.6.5 ar
- 2.7 内核发言：倾听来自内核的消息
  - 2.7.1 printk()
  - 2.7.2 dmesg
  - 2.7.3 /var/log/messages
- 2.8 其他奥秘
  - 2.8.1 \_\_init
  - 2.8.2 likely()和unlikely()
  - 2.8.3 IS\_ERR 和PTR\_ERR
  - 2.8.4 通告程序链
- 2.9 小结
  - 2.9.1 项目：Hellomod
  - 2.9.2 第一步：构造Linux 模块的框架
  - 2.9.3 第二步：编译模块
  - 2.9.4 第三步：运行代码
- 2.10 习题

## 第3章 进程：程序执行的基本模型

- 3.1 程序
- 3.2 进程描述符
  - 3.2.1 与进程属性相关的字段
  - 3.2.2 与调度相关的字段
  - 3.2.3 涉及进程间相互关系的字段
  - 3.2.4 与进程信任状相关的字段
  - 3.2.5 与进程权能相关的字段
  - 3.2.6 与进程限制相关的字段
  - 3.2.7 与文件系统及地址空间相关的字段
- 3.3 进程的创建：系统调用fork()、vfork 和clone()
  - 3.3.1 fork()函数
  - 3.3.2 vfork()函数
  - 3.3.3 clone()函数
  - 3.3.4 do\_fork()函数
- 3.4 进程的生命周期
  - 3.4.1 进程的状态
  - 3.4.2 进程状态的转换
- 3.5 进程的终止

## &lt;&lt;Linux内核编程&gt;&gt;

- 3.5.1 sys\_exit() 函数
- 3.5.2 do\_exit()函数
- 3.5.3 通知父进程和sys\_wait4()
- 3.6 了解进程的动态：调度程序的基本构架
  - 3.6.1 基本结构
  - 3.6.2 从等待中醒来或者激活
- 3.7 等待队列
  - 3.7.1 添加到等待队列
  - 3.7.2 等待事件
  - 3.7.3 唤醒进程
- 3.8 异步执行流程
  - 3.8.1 异常
  - 3.8.2 中断
- 3.9 小结
  - 3.9.1 项目：系统变量current
  - 3.9.2 项目源码
  - 3.9.3 运行代码
- 3.10 习题
- 第4章 内存管理
  - 4.1 页
    - 4.2 内存管理区
      - 4.2.1 内存管理区描述符
      - 4.2.2 内存管理区操作辅助函数
    - 4.3 页面
      - 4.3.1 请求页面的函数
      - 4.3.2 释放页面的函数
      - 4.3.3 伙伴系统
    - 4.4 Slab 分配器
      - 4.4.1 缓存描述符
      - 4.4.2 通用缓存描述符
      - 4.4.3 Slab 描述符
    - 4.5 Slab 分配器的生命周期
      - 4.5.1 与Slab 分配器有关的全局变量
      - 4.5.2 创建缓存
      - 4.5.3 创建slab 与cache\_grow()
      - 4.5.4 Slab 的销毁：退还内存与kmem\_cache\_destroy()
    - 4.6 内存请求路径
      - 4.6.1 kmalloc()
      - 4.6.2 kmem\_cache\_alloc()
    - 4.7 Linux 进程的内存结构
      - 4.7.1 mm\_struct
      - 4.7.2 vm\_area\_struct
    - 4.8 进程映像的分布及线性地址空间
    - 4.9 页表
    - 4.10 缺页
      - 4.10.1 x86 缺页异常
      - 4.10.2 缺页处理程序

## &lt;&lt;Linux内核编程&gt;&gt;

4.10.3 PowerPC 缺页异常

4.11 小结

4.12 项目：进程内存映射

4.13 习题

## 第5章 输入/输出

5.1 总线、桥、端口和接口的硬件实现

5.2 设备

5.2.1 块设备概述

5.2.2 请求队列和I/O 调度

5.2.3 示例：“通用”块设备驱动程序

5.2.4 设备操作

5.2.5 字符设备

5.2.6 网络设备

5.2.7 时钟设备

5.2.8 终端设备

5.2.9 直接存储器存取

5.3 小结

5.4 项目：创建并口驱动程序

5.4.1 并口的硬件

5.4.2 运行在并口上的软件

5.5 习题

## 第6章 文件系统

6.1 文件系统的基本概念

6.1.1 文件和文件名

6.1.2 文件类型

6.1.3 文件的附加属性

6.1.4 目录和路径名

6.1.5 文件操作

6.1.6 文件描述符

6.1.7 磁盘块、磁盘分区以及实现

6.1.8 性能

6.2 Linux 虚拟文件系统

6.2.1 VFS 的数据结构

6.2.2 全局链表和局部链表的引用

6.3 与VFS 相关的结构

6.3.1 fs\_struct 结构

6.3.2 files\_struct 结构

6.4 页缓存

6.4.1 address\_space 结构

6.4.2 buffer\_head 结构

6.5 VFS 的系统调用和文件系统层

6.5.1 open()

6.5.2 close()

6.5.3 read()

6.5.4 write()

6.6 小结

## &lt;&lt;Linux内核编程&gt;&gt;

## 6.7 习题

## 第7章 进程调度和内核同步

## 7.1 Linux 的调度程序

## 7.1.1 选择下一个进程

## 7.1.2 上下文切换

## 7.1.3 让出CPU

## 7.2 内核抢占

## 7.2.1 显式内核抢占

## 7.2.2 隐式用户抢占

## 7.2.3 隐式内核抢占

## 7.3 自旋锁和信号量

## 7.4 系统时钟：关于时间和定时器

## 7.4.1 实时时钟：现在几点了

## 7.4.2 读取PPC 的实时时钟

## 7.4.3 读取x86 的实时时钟

## 7.5 小结

## 7.6 习题

## 第8章 内核引导

## 8.1 BIOS 和Open Firmware

## 8.2 引导加载程序

## 8.2.1 GRUB

## 8.2.2 LILO

## 8.2.3 PowerPC 和Yaboot

## 8.3 与体系结构相关的内存初始化

## 8.3.1 PowerPC 的硬件内存管理

## 8.3.2 基于Intel x86 体系结构的硬件内存管理

## 8.3.3 PowerPC 和x86 的代码汇集

## 8.4 原始的RAM 盘

## 8.5 开始：start\_kernel()

## 8.5.1 调用lock\_kernel()

## 8.5.2 调用page\_address\_init()

## 8.5.3 调用printk(linux\_banner)

## 8.5.4 调用setup\_arch

## 8.5.5 调用setup\_per\_cpu\_areas()

## 8.5.6 调用smp\_prepare\_boot\_cpu()

## 8.5.7 调用sched\_init()

## 8.5.8 调用build\_all\_zonelists()

## 8.5.9 调用page\_alloc\_init

## 8.5.10 调用parse\_args()

## 8.5.11 调用trap\_init()

## 8.5.12 调用rcu\_init()

## 8.5.13 调用init\_IRQ()

## 8.5.14 调用softirq\_init()

## 8.5.15 调用time\_init()

## 8.5.16 调用console\_init()

## 8.5.17 调用profile\_init()

## &lt;&lt;Linux内核编程&gt;&gt;

- 8.5.18 调用local\_irq\_enable()
- 8.5.19 配置initrd
- 8.5.20 调用mem\_init()
- 8.5.21 调用late\_time\_init()
- 8.5.22 调用calibrate\_delay()
- 8.5.23 调用pgtable\_cache\_init()
- 8.5.24 调用buffer\_init()
- 8.5.25 调用security\_scaffolding\_startup()
- 8.5.26 调用vfs\_caches\_init()
- 8.5.27 调用radix\_tree\_init()
- 8.5.28 调用signal\_init()
- 8.5.29 调用page\_writeback\_init()
- 8.5.30 调用proc\_root\_init()
- 8.5.31 调用init\_idle()
- 8.5.32 调用rest\_init()
- 8.6 init 线程 ( 或进程1 )
- 8.7 小结
- 8.8 习题

## 第9章 构建Linux 内核

- 9.1 工具链
  - 9.1.1 编译程序
  - 9.1.2 交叉编译
  - 9.1.3 链接程序
  - 9.1.4 ELF 二进制目标文件
- 9.2 内核源代码的构建
  - 9.2.1 解释源代码
  - 9.2.2 构建内核映像
- 9.3 小结
- 9.4 习题

## 第10章 向内核添加代码

- 10.1 浏览源代码
  - 10.11 熟悉文件系统
  - 10.12 filp 和fops
  - 10.13 用户空间和内核空间
  - 10.14 等待队列
  - 10.15 工作队列及中断
  - 10.16 系统调用
  - 10.17 其他类型的驱动程序
  - 10.18 设备模型和sysfs 文件系统
- 10.2 编写代码
  - 10.2.1 设备基础
  - 10.2.2 符号输出
  - 10.2.3 IOCTL
  - 10.2.4 轮询与中断
  - 10.2.5 工作队列和tasklet

10.2.6 增加系统调用的代码

10.3 构建和调试

10.4 小结

10.5 习题

## <<Linux内核编程>>

### 媒体关注与评论

“本书详细比较了x86和PPC体系结构下的汇编程序，并介绍了分析工具。非常不错！”

——亚马逊读者评论

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>