

Linux 与自由软件资源丛书

Linux 编程白皮书

David A. Rusling 等著

朱珂 涂二靓 等译



机械工业出版社
China Machine Press

本书对Linux操作系统及其编程作了整体的介绍,以支持用于开发软件的公开源码模型。对内存管理、进程及其通信机制、PCI、内核模块编程及内核系统结构作了详细的解释,且附有很多程序代码实例。对深入研究Linux下的编程有很大的帮助。

David A. Rusling et al: Linux Programming White Papers

本书遵从GNU通用公共许可证。详细信息请参照第六部分的附录B。

The Linux Kernel

© 1996-1998 David A. Rusling

This book is freely distributable; you may copy and redistribute it under certain conditions.

Please refer to the copyright and distribution statement.

Version 0.8-3

David A. Rusling

The Linux Kernel Module Programming Guide

© 1999 Ori Pomerantz

The Linux Programmer's Guide

© 1994, 1995 by Sven Goldt

Sven Goldt, Sachsendamm 47b, 10829 Berlin, Germany (goldt@math.tu-berlin.de)

Chapter 6 is © 1995 by Scott Burkett (scottb@lnrNet.net)

Chapter 8 is © 1994, 1995 by Sven van der Meer (vdmeer@cs.tu-berlin.de)

Chapter 10 is © 1994, 1995 by Matt Welsh (mdw@cs.cornell.edu)

Special thanks go to John D. Harper (jharper@uiuc.edu) for proofreading this guide.

Permission to reproduce this document in whole or in part is subject to the following conditions:

- The copyright notice remains intact and is included.
- If you make money with it the authors want a share.
- The authors are not responsible for any harm that might arise by the use of it.

图书在版编目(CIP)数据

Linux 编程白皮书 / 路丝林(Rusling,D.A.)等著; 朱珂等译. - 北京: 机械工业出版社, 2000.4

(Linux 与自由软件资源丛书)

书名原文: Linux Programming White Papers

ISBN 7-111-07804-7

I. L... II. ①路... ②朱... III. 操作系统(软件). Linux IV. TP316.89

中国版本图书馆CIP数据核字(2000)第15351号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 李云静 周桦

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2000年4月第1版第1次印刷

787mm×1092mm 1/16·24.5印张

印数: 0 001-7 000册

定价: 39.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

雷 蒙 序

长期以来有一种观念，认为计算机界的程序员一般都厌恶编写文档。这是不正确的……对于他们真正关心的程序，许多程序员会非常高兴地为之编写文档。而最好的例子就是读者正在阅读的这本书。

Linux是众多开发它的程序员的至爱。它取得了令人震惊的成功，在三年时间内，从一个无名之辈成为一个主要的竞争者，这说明在推动高质量创造性工作方面，爱的力量比金钱的力量要强大得多。但是，出版图书和光盘又确实需要花钱，所以Linux世界正在逐渐与市场融合，从而给出版业带来了一股Linux热潮，这本书是这种趋势的又一个例证。

Coriolis Group公司正在扩展与Linux文档工程小组合作的方法，以便帮助Linux开发人员做好他们的开发工作(编码以及撰写技术参考文献)，而Linux用户也可以得到质量更佳的在线文档。所有的LDP文档将来会继续在支持自由发布的公开内容许可证下发行，而Coriolis的努力也将回报整个社会。

Eric Raymond

简介

Linux文档页面(LDP)站点上 (<http://metalab.unc.edu/LDP/>) 共有八本在线书籍,《Linux编程白皮书》是其中五本书籍的印刷版本。Coriolis Group公司正在寻求各种方法,以便支持Linux以及用于开发软件的公开源码模型。我们认为,要做到这一点,可以把LDP分为两卷出版。这一卷命名为《Linux编程白皮书》,而另一卷的重点则是管理方面的问题。请一定要访问LDP站点,以便得到LDP提供的其他Linux资源。作为这个简介的一部分,读者会读到LDP公告。请阅读它,它会帮助读者更深入地了解LDP。

根据这份资料中作者的有关版权声明,我们得以出版这本书的印刷版本。这也意味着本书中所含资料都可以自由地再发行,只要保留原作者的版权声明即可。请阅读第六部分的附录A《Linux文档工程拷贝许可证》,它清楚地指出了这些版权规定。此外,请阅读第六部分的附录B《GNU公共许可证》,Linux遵守该许可证。如果读者还有什么问题,可以访问GNU工程小组的网址www.gnu.org。

在阅读本书的时候,读者会发现其中有些资料已经过时,而有的例题则没有提供,这是因为在使用LDP时要受到版权声明的制约,我们不能擅自修改作者的资料,甚至改正错误和遗漏也不行。然而,在发现遗漏和错误的地方,我们都作出了相应的标记。请一定要访问在线LDP,以便获取本书中的文档的更新版本。

Coriolis Group一直在寻求进一步支持Linux的方法。如果你有什么建议,可以帮助我们做到这一点,且愿意把它提供给我们,可直接与我们联系,邮箱地址是ctp@coriolis.com。我们期待你的评价。

Coriolis Group公司

Linux文档工程小组“公告”

最近一次修改是由Michael k.Johnson于1998年9月21日完成的。

本文件介绍了Linux文档工程小组的目标和现状，包括工程小组的名称、志愿者、FTP站点等等。

概述

Linux文档工程小组致力于为Linux操作系统开发可靠的好文档。LDP的总目标是关注所有Linux文档事务，从在线文档(man帮助信息页、texinfo文档等等)到打印的手册。内容包括Linux的安装、使用和运行。LDP基本上是由志愿者组成的松散的队伍，没有什么中心组织，任何有兴趣帮忙的人都可以参加。我们认为在一起工作，并认定文档的方向和范围是很重要的，这样可以避免出现重复劳动——两个人都写Linux同一方面的书必将导致时间和精力浪费。

参加

请发送电子邮件到linux_howto@metalab.unc.edu。

当然，读者可能需要了解当前LDP工程小组的工作，看自己是否有兴趣参加，请阅读下面内容。

当前工程

如果想得到我们当前进行的工程的一个列表，可以访问LDP的网页<http://metalab.unc.edu/LDP/>。如果想参加这些工程中的其中一个，最好的方法是下载该手册的当前版本，并把你的修改、编辑和建议发送给该工程的负责人。读者可能希望在发送修改意见以前与作者联系，这样读者就可以与作者一起工作了。

LDP文献的FTP站点

LDP文档都可以在metalab.unc.edu上的目录/pub/Linux/docs下找到。LDP手册在目录/pub/Linux/docs/LDP下，而HOWTO和其他文档则位于/pub/Linux/docs/HOWTO下。

文档的约定

下面列举了LDP手册当前所使用的一些约定。如果读者有兴趣用其他的约定编写另外的手册，请首先把你的计划告诉我们。

man pages(帮助页面)——在线手册的Unix标准——它们是使用Unix标准的nroff man(或者BSD mdoc)宏创建的。

guides(指南)——由LDP编写的完整的书——原来是用LaTeX完成的，它们的主要目标也是成为印刷文档。然而，指南的作者现在倾向于SGML格式，逐渐开始使用DocBook DTD，因为它让这些作者可以创建更多的不同类型的输出，可以是印刷的和在线的，如果用户使用LaTeX，我们提供了一个样式文件，可以用来使用户印刷的书与其他LDP文档保持一致，我们建议用户使用它。

HOWTO文档都必须是SGML格式的。当前，它们使用linuxdoc DTD，它是非常简单的，目前人们正计划在一段时间内逐渐使用DocBook DTD来替换它。

LDP文档必须是自由的、可重发布的、重新发布无需向作者支付任何费用。虽然并不要求文档是可修改的，但是鼓励这么做。读者在编写LDP文档时可以满足这个要求，或者也可以使用以前准备好的许可证。LDP提供了样板许可证。可供用户使用，有些人喜欢使用GPL，而另一些人则使用自己的许可证。

每个手册的版权都必须加上第一作者或工程负责人的名称。“Linux文档工程小组”不是一个正规的组织，不能用在这些文档的版权中。

译者序

Linux是当前最流行的操作系统之一。它具有高效、健壮、安全以及功能齐全等特性，因而赢得了人们的青睐。它是一个完全免费的Unix操作系统，来自世界各地的志愿者为这个充满魅力的操作系统的发展贡献着自己的才能。无论是初学者还是资深的软件工程师、都有机会对这个操作系统进行改进和完善。可以说，Linux的发展凝聚了世界各地无数开发人员的大量心血，体现了一种信息世界的共建、共享和共荣的精神。

因为Linux是一个完全公开的操作系统，每个人都可以得到它的源码，对它进行研究、扩充、传播和使用。这使得许多人的天才有了用武之地。目前国内，自学成为Linux专家几乎成了年轻人最大的梦想之一。

本书分为6个部分，每部分独立成篇。全书重点介绍Linux内核模块编程以及应用程序编程，帮助读者掌握内存管理、模块编程、进程间通信、中断处理、I/O端口编程的有关知识；并帮助读者理解/proc文件系统、对称多处理、设备驱动程序、目录树以及libc服务包等概念。与其他Linux编程书籍不同的是，本书详细介绍了Linux的系统结构，使读者能深入了解Linux内核的构成以及各个组成部分之间的相互关系。另外，本书对于进程间通信机制的介绍是目前所有Linux编程书籍中最详细和完备的。

本书第一部分由朱珂、曹宏佳、张小哲、杨华坤、窦文、尹刚、张杰翻译，其余部分由涂二砚组织翻译，前导工作室的全体人员参加了本书的翻译。因为书中涉及了大量的新技术，加上译者水平有限，错误之处在所难免。欢迎读者指正。

译者
2000.1

目 录

雷蒙序

简介

Linux文档工程小组“公告”

译者序

第一部分 Linux内核

前言

第1章 硬件基础与软件基础6

1.1 硬件基础6

1.1.1 CPU7

1.1.2 存储器8

1.1.3 总线8

1.1.4 控制器和外设8

1.1.5 地址空间9

1.1.6 时钟9

1.2 软件基础9

1.2.1 计算机语言9

1.2.2 什么是操作系统11

1.2.3 内核数据结构13

第2章 内存管理15

2.1 虚拟内存抽象模型15

2.1.1 请求调页17

2.1.2 交换17

2.1.3 共享虚拟内存18

2.1.4 物理寻址模式和虚拟寻址模式18

2.1.5 访问控制18

2.2 高速缓存19

2.3 Linux页表20

2.4 页分配和回收21

2.4.1 页分配22

2.4.2 页回收22

2.5 内存映射22

2.6 请求调页23

2.7 Linux页缓存24

2.8 页换出和淘汰25

2.8.1 减少缓冲区和页缓存大小25

2.8.2 换出System V共享内存页26

2.8.3 换出和淘汰页27

2.9 交换缓存27

2.10 页换入28

第3章 进程29

3.1 Linux进程29

3.2 标识符31

3.3 调度32

3.4 文件34

3.5 虚拟内存35

3.6 创建进程36

3.7 时间和定时器37

3.8 执行程序38

3.8.1 ELF39

3.8.2 脚本文件40

第4章 进程间通信机制41

4.1 信号机制41

4.2 管道42

4.3 套接字44

4.3.1 System V的进程间通信机制44

4.3.2 消息队列44

4.3.3 信号量45

4.3.4 共享存储区47

第5章 PCI49

5.1 PCI的地址空间49

5.2 PCI配置头50

5.3 PCI的I/O和存储地址空间51

5.4 PCI-ISA桥51

5.5 PCI-PCI桥51

5.5.1 PCI-PCI桥: PCI I/O和存储地址
空间的窗口51

5.5.2 PCI-PCI桥: PCI配置周期和PCI
总线编号52

5.6 Linux PCI初始化	53	7.5 特殊设备文件	91
5.6.1 Linux内核PCI数据结构	53	第8章 网络	92
5.6.2 PCI设备驱动程序	53	8.1 TCP/IP网络概述	92
5.6.3 PCI的BIOS函数	56	8.2 Linux中的TCP/IP网络层次结构	95
5.6.4 PCI修正过程	57	8.3 BSD套接字接口	96
第6章 中断处理与设备驱动程序	60	8.4 INET的套接字层	97
6.1 中断与中断处理	60	8.4.1 创建BSD套接字	98
6.1.1 可编程中断控制器	61	8.4.2 为INET BSD Socket绑定地址	99
6.1.2 初始化中断处理数据结构	61	8.4.3 建立INET BSD Socket连接	99
6.1.3 中断处理	62	8.4.4 INET BSD Socket侦听	100
6.2 设备驱动程序	63	8.4.5 接受连接请求	100
6.2.1 测试与中断	64	8.5 IP层	100
6.2.2 直接存储器访问(DMA)	65	8.5.1 套接字缓冲区	100
6.2.3 存储器	66	8.5.2 接收IP报文	101
6.2.4 设备驱动程序与内核的接口	66	8.5.3 发送IP报文	102
6.2.5 硬盘	69	8.5.4 数据分片	102
6.2.6 网络设备	74	8.6 地址解析协议	103
第7章 文件系统	77	8.7 IP路由	104
7.1 第二个扩展文件系统EXT2	78	第9章 内核机制与模块	107
7.1.1 EXT2系统的inode节点	79	9.1 内核机制	107
7.1.2 EXT2系统的超级块	80	9.1.1 Bottom Half控制	107
7.1.3 EXT2系统的组描述符	80	9.1.2 任务队列	108
7.1.4 EXT2系统的目录	81	9.1.3 定时器	109
7.1.5 在EXT2文件系统中查找文件	81	9.1.4 等待队列	110
7.1.6 在EXT2文件系统中改变文件 的大小	82	9.1.5 自旋锁	110
7.2 虚拟文件系统	83	9.1.6 信号量	110
7.2.1 VFS文件系统的超级块	84	9.2 模块	111
7.2.2 VFS文件系统的inode节点	84	9.2.1 模块载入	112
7.2.3 注册文件系统	85	9.2.2 模块卸载	113
7.2.4 装配文件系统	85	第10章 处理器	115
7.2.5 在虚拟文件系统中查找文件	87	10.1 X86	115
7.2.6 卸载文件系统	87	10.2 ARM	115
7.2.7 VFS文件系统的inode缓存	87	10.3 Alpha AXP处理器	115
7.2.8 目录缓存	88	第11章 Linux内核源代码	117
7.3 缓冲区缓存	88	11.1 怎样得到Linux内核源码	117
7.3.1 bdfush内核守护进程	90	11.2 内核源码的编排	117
7.3.2 update进程	90	11.3 从何处看起	118
7.4 /proc文件系统	91	第12章 Linux数据结构	120
		附录A 有用的Web和FTP站点	138

附录B 词汇表	139
---------------	-----

第二部分 Linux内核模块编程指南

致谢

前言

第1章 Hello, World	145
1.1 内核模块的Makefiles文件	146
1.2 多重文件内核模块	147
第2章 字符设备文件	149
第3章 /proc文件系统	158
第4章 把/proc用于输入	162
第5章 把设备文件用于输入	170
第6章 启动参数	182
第7章 系统调用	185
第8章 阻塞处理	190
第9章 替换printk	199
第10章 任务调度	202
第11章 中断处理程序	207
第12章 对称多处理	211
第13章 常见错误	212
附录A 2.0和2.2之间的差异	213
附录B 其他资源	214
附录C 给出你的评价	215

第三部分 Linux程序员指南

第1章 Linux操作系统	219
第2章 Linux内核	220
第3章 Linux libc包	221
第4章 系统调用	222
第5章 “瑞士军刀”: ioctl	223
第6章 Linux进程间通信	224
6.1 介绍	224
6.2 半双工Unix管道	224
6.2.1 基本概念	224
6.2.2 用C语言创建管道	225
6.2.3 简便方法	229
6.2.4 管道的原子操作	233
6.2.5 关于半双工管道需要注意的几个问题	233

6.3 命名管道	234
6.3.1 基本概念	234
6.3.2 创建FIFO	234
6.3.3 FIFO操作	235
6.3.4 FIFO上的阻塞动作	236
6.3.5 SIGPIPE信号	237
6.4 系统V IPC	237
6.4.1 基本概念	237
6.4.2 消息队列	239
6.4.3 信号量	251
6.4.4 semtool: 交互式信号量操作程序	260
6.4.5 共享内存	267
第7章 声音编程	274
7.1 内部扬声器编程	274
7.2 声卡编程	274
第8章 字符单元图形	276
8.1 libc中的I/O函数	277
8.1.1 格式化输出	277
8.1.2 格式化输入	278
8.2 termcap库	278
8.2.1 前言	278
8.2.2 获得终端描述	279
8.2.3 查看终端描述	279
8.2.4 termcap权能	280
8.3 Ncurses简介	280
8.4 初始化	282
8.5 窗口	283
8.6 输出	284
8.6.1 格式化输出	285
8.6.2 插入字符/行	286
8.6.3 删除字符/行	286
8.6.4 方框和直线	287
8.6.5 背景字符	287
8.7 输入	288
8.8 选项	289
8.8.1 输出选项	289
8.8.2 输入选项	290
8.8.3 终端属性	291

8.8.4 使用选项	291
8.9 更新终端	292
8.10 视频属性与颜色	293
8.11 光标和屏幕坐标	294
8.12 滚动	294
8.13 小键盘	295
8.14 软标签	295
8.15 杂项	295
8.16 低级访问	296
8.17 屏幕转储	296
8.18 Termcap模拟	296
8.19 Terminfo函数	296
8.20 调试函数	297
8.21 Terminfo权能	297
8.21.1 布尔型权能	297
8.21.2 数值型权能	298
8.21.3 字符串型权能	299
8.22 [N]Curses函数概述	306
第9章 I/O端口编程	307
9.1 鼠标编程	307
9.2 调制解调器编程	308
9.3 打印机编程	308
9.4 游戏杆编程	308
第10章 把应用程序移植到Linux上	309
10.1 介绍	309
10.2 信号处理	309
10.2.1 SVR4、BSD和POSIX.1下 的信号	310
10.2.2 Linux信号选项	310
10.2.3 Linux下的信号	310
10.2.4 Linux支持的信号	311
10.3 终端I/O	311
10.4 进程信息和控制	311
10.4.1 kvm过程	312
10.4.2 ptrace和/proc文件系统	312
10.4.3 Linux下的进程控制	312
10.5 可移植条件编译	313
10.6 补充说明	314
附录 以字母顺序排列的系统调用	315

第四部分 Linux内核概念系统结构

摘要	323
前言	324
第1章 系统结构	325
1.1 系统概述	325
1.2 内核的目标	325
1.3 内核结构的概述	325
1.4 支持多个开发人员	327
1.5 系统数据结构	328
1.5.1 任务列表	328
1.5.2 内存映射	328
1.5.3 索引节点	328
1.5.4 数据连接	329
第2章 子系统的系统结构	330
2.1 进程调度程序系统结构	330
2.1.1 目标	330
2.1.2 模块	330
2.1.3 数据表达	331
2.1.4 依赖性、数据流和控制流	331
2.2 内存管理程序系统结构	331
2.2.1 目标	331
2.2.2 模块	331
2.2.3 数据表示	331
2.2.4 数据流、控制流和依赖性	332
2.3 虚拟文件系统系统结构	333
2.3.1 目标	333
2.3.2 模块	333
2.3.3 数据表示	333
2.3.4 数据流、控制流和依赖性	334
2.4 网络接口系统结构	334
2.4.1 目标	334
2.4.2 模块	334
2.4.3 数据表示	335
2.4.4 数据流、控制流和依赖性	335
2.5 进程间通信系统结构	335
第3章 结论	336
附录A 术语定义	337
附录B 参考文献	338

第五部分 Linux内核具体系统结构

摘要	341
第1章 前言	342
1.1 目标	342
1.2 Linux介绍	342
1.3 软件系统结构的背景知识	342
1.4 方法与途径	343
1.5 适用本书的读者	344
1.6 本部分的章节安排	344
第2章 系统结构	345
第3章 子系统结构	346
3.1 进程调度程序	346
3.1.1 目标	346
3.1.2 外部接口	346
3.1.3 子系统描述	346
3.1.4 数据结构	347
3.1.5 子系统结构	348
3.1.6 子系统依赖性	348
3.2 内存管理程序	348
3.2.1 目标	348
3.2.2 外部接口	349
3.2.3 子系统描述	349
3.2.4 数据结构	350
3.2.5 子系统结构	350
3.2.6 子系统依赖性	351
3.3 虚拟文件系统	352
3.3.1 目标	352
3.3.2 外部接口	352

3.3.3 子系统描述	353
3.3.4 设备驱动程序	353
3.3.5 逻辑文件系统	354
3.3.6 模块	354
3.3.7 数据结构	355
3.3.8 子系统结构	355
3.3.9 子系统依赖性	355
3.4 进程间通信	355
3.4.1 目标	355
3.4.2 外部接口	357
3.4.3 子系统描述	357
3.4.4 数据结构	358
3.4.5 子系统结构	359
3.4.6 子系统依赖性	359
3.5 网络接口	360
3.5.1 目标	360
3.5.2 外部接口	361
3.5.3 子系统描述	361
3.5.4 数据结构	362
3.5.5 子系统结构	363
3.5.6 子系统依赖性	363
第4章 结论	365
附录A 术语定义	366
附录B 参考文献	368

第六部分 附录

附录A Linux文档工程拷贝许可证	372
附录B GNU通用公共许可证	374

第一部分 Linux 内核

David A. Rusling 著

This book is for Linux enthusiasts who want to know how the Linux Kernel works. It is not an internals manual. Rather it describes the principles and mechanisms that Linux uses, how and why the Linux kernel works the way that it does.

Linux is a moving target; this book is based upon the current, stable, 2.0.33 sources as those are what most individuals and companies are now using.

This book is freely distributable, you may copy and redistribute it under certain conditions. Please refer to the copyright and distribution statement.

Version 0.8-3

David A. Rusling (david.rusling@arm.com)

前言

Linux 是互联网(Internet)的产物。作为一名学生的业余爱好项目,它已变得比其他任何可以免费获得的操作系统更流行。对许多人来说, Linux 是个谜。免费的东西怎么会有用?在被屈指可数的几个大型软件公司统治的世界里,一群“黑客”写出的东西怎么会有希望竞争获胜呢?由来自世界上不同国家的许多不同的人写出的软件怎么会是稳定而高效的呢?然而它确实稳定又高效,并在竞争中获得胜利。许多大学和研究机构在它们的日常计算中使用它。人们在家里的PC上运行它;并且大多数公司正在某些地方使用它。Linux被用来浏览网页、建造网站、写论文、发电子邮件,以及像在一般的计算机上一样玩游戏。重要的是Linux不是一个玩具,它是一个全面开发的和专业化写出的操作系统,被全世界的爱好者所使用。

Linux的发展可以追溯到UNIX的起源。1969年,贝尔实验室研究组的Ken Thompson开始用一台闲置的PDP-7试着建立一个多用户、多任务的操作系统。很快Dennis Richie加入了他的工作,他们俩和研究组中其他成员一起写出UNIX的早期版本。Richie 深受早期的一个工程MULTICS的影响,UNIX的这个名字本身就与MULTICS有关。UNIX早期版本是用汇编代码写的,但是它的第3版是用一个新的语言C重写的。C是Richie专门为写操作系统而设计和开发的语言。这次重写使得UNIX可以被移植到DIGITAL公司那时生产的能力更强大的PDP-11/45和11/70计算机上。之后,UNIX从实验室中走出并进入主流计算,并且很快大部分主要计算机生产厂家开始开发出他们自己的版本。

Linux是针对简单需求的一个解决方案。当时Linus——Linux的作者和规范维护者,唯一能够买得起的软件就是Minix。Minix是一个被广泛用于帮助教学的、简单的类UNIX操作系统。Linus并未被它的特征打动,他的目的是写自己的软件。他拿UNIX作为模型,因为那是他在每天的学习生活中所熟悉的一种操作系统。他开始写时只有一台Intel 386PC。Linus对于迅速的进展感到很激动,就把自己的成果通过正在开始出现的计算机国际互联网提供给其他学生。那时国际互联网还主要应用于学术群体。其他人看到这个软件并开始进行开发。这个新软件的大部分就是这些开发人员自己对某个问题的解决方案。不久, Linux成为一个操作系统。很有必要说明的一点是Linux不含任何UNIX代码,它是基于发布的POSIX标准而重写的。Linux的建造使用了大量的自由软件基金会提供的GNU(是GNU的而不是UNIX的)软件。

大量Linux使用者用它来写应用程序或运行别人写的应用程序。许多Linux使用者贪婪地阅读“HOWTO”^①,当系统的某部分被正确配置的时候他们会感到成功的喜悦,否则也会因失败而感到沮丧。一部分人甚至写一些设备驱动程序并提供内核补丁给Linus Torvalds——Linux内核的创造者和维护者。Linus接受来自任何地方、任何人的对内核源代码的增补和修改。这听起来可能有点像无政府的方法,但Linus实施严格的质量控制,并且亲自把新代码归并到内核中。

大多数的Linux使用者并不看Linux操作系统如何工作、它是怎样组合在一起的。这很不足取,因为研究Linux可学习有关操作系统如何工作的更多知识。它不仅写得很好,并且所有

① 正如它的名字一样,是一份描述如何去做某件事情的文档。有好多为Linux写的这种文档,并且全都很有用处。

的源代码都可以免费得到。这是因为尽管作者们保留他们软件的版权，但却允许源代码在自由软件基金会的GNU公共许可下免费发布。乍一看源代码可能会感到很迷惑：你可能看到像Kernel、mm、net这样的目录，但它们包含什么，代码又是如何工作的呢？你所需要的是对Linux的目标和整体结构的更深理解。简短地说，这就是本书的目的：给出一个对Linux操作系统如何工作的清晰解释。提供一个脑海中的模型，使得当你从一个地方到另一个地方拷贝文件或阅读电子邮件的时候，能够想象出系统内部发生了什么。

Alpha AXP只是可以运行Linux的众多硬件平台之一。大部分Linux内核运行在基于Intel处理器的系统上，但越来越多的非Intel Linux系统变得更容易得到。这其中包括Alpha AXP、MIPS、Sparc和PowerPC。可以基于以上任何一种平台来写这本书，但本书背景知识和技术经验是在Alpha AXP上的，这就是本书有时用该硬件作为例子来展开某些关键点的原因。不过，大约95%的Linux内核源代码对于它所运行的所有硬件平台是相同的，也就是说，大约本书95%的内容是与Linux内核的机器无关的。

关于读者

本书不对读者的知识和经验做任何假设。我们相信在需要的时候对一个主题的兴趣能够激励人自学。尽管如此，对计算机，最好是对PC有一定程度的熟悉，将有助于读者从本材料中获得真正的好处。同样的，对C语言有所了解也会对读者有帮助。

本书的组织

本书并不想被用作Linux的一本内部手册。相反，它是一本对操作系统，特别是针对Linux的整体介绍。本书的各章都遵循“从一般到特殊”的原则。它们先给出所描述的核心子系统的一个概述，然后再进入到复杂的细节。

我们有意地没有用像“routine_X()调用了routine_Y()，在其中将bar数据结构的foo字段递增”这样的术语描述内核的算法，即它完成工作的方法。你可以通过读代码来发现这些。当需要理解一段代码或要给别人描述它的时候，可先画出它的数据结构。所以，本书也详细描述了有关的内核数据结构及它们之间的关系。

每一章都是完全独立的，就像它们所描述的Linux内核子系统一样。然而有时候会有些联系，比如如果不理解虚拟内存如何工作就无法描述进程。

第1章首先给出了当代PC的一个简介。一个操作系统必须与作为它基础的硬件系统紧密工作在一起。操作系统需要只有硬件能提供的特定服务。为了完全理解Linux操作系统，你需要了解它下层硬件的基础。接着介绍了基本的软件原理，并涉及了汇编语言和C语言。它介绍了用来建造一个像Linux这样的操作系统所使用的工具并给出了操作系统目标和功能的概述。

第2章描述了Linux处理物理和虚拟内存的方法。

进程之间和进程与内核之间要相互通信以协调它们的行为。Linux支持几种进程间通信(IPC)机制。信号和管道是其中的两个，但Linux也支持那些以它们最早出现的UNIX发布来命名的System V中的IPC机制。这些进程间通信机制在第4章描述。

外围部件互连(PCI)标准作为低价格高性能的PC数据总线已经牢固树立起来。第5章描述Linux怎样初始化和使用系统中的PCI总线和设备。

第6章首先介绍Linux如何处理中断。尽管内核有通用的处理中断的机制和接口，但有些中断处理细节是硬件和体系结构相关的。接着介绍Linux如何控制系统中的物理设备。

第7章介绍Linux如何维护在它所支持的文件系统中的文件，该章描述了虚拟文件系统(VFS)及Linux内核的真实文件系统是如何被支持的。

网络和Linux几乎是同义术语。在相当程度上Linux是国际互联网或万维网(WWW)的产物。其开发者和使用者使用网络来交换意见信息和代码，Linux本身经常用来支持单位的网络需要。第8章描述了Linux如何支持众所周知的网络协议TCP/IP。

第9章涉及一些通用任务和机制，Linux内核需要支持它们以使得内核中其他部分高效地配合在一起工作；描述Linux内核如何在需要的时候动态装载一些功能，比如文件系统。

其余章节还介绍了Linux数据结构、处理器，有用的Web和FTP站点及Linux内核源码等内容。

本书假设你想看源代码本身，并且所有的文件引用都是相对于usr/src/linux的。拿foo/bar.c为例，完整的文件名应该是usr/src/linux/foo/bar.c。如果你在运行Linux，那么查看源代码是很值得的做法，你可以把本书作为理解源代码和其众多数据结构的向导。