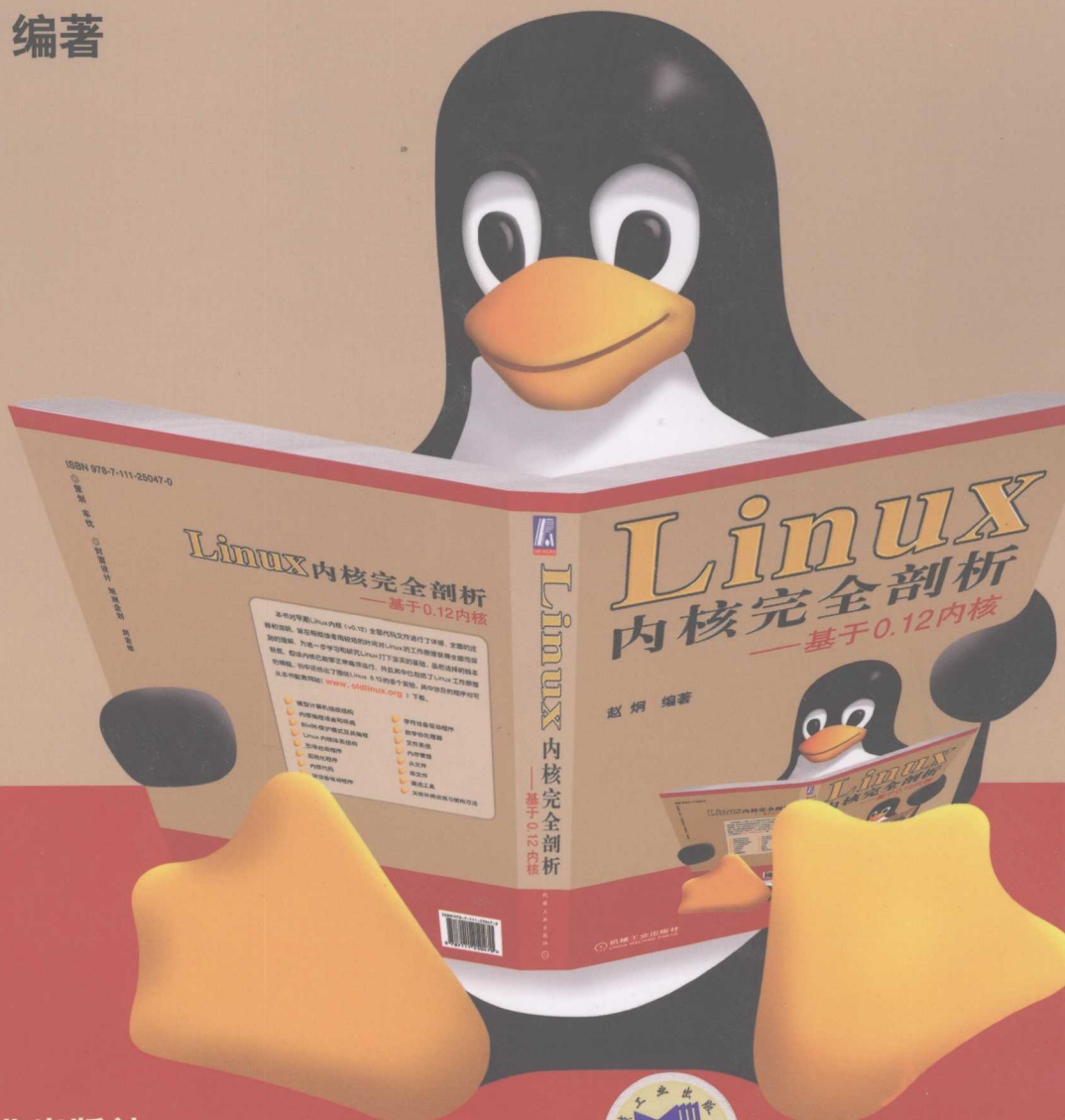
Linux 内核完全剖析

——基于0.12内核

赵 炯 编著



机械工业出版社
CHINA MACHINE PRESS

Linux 内核完全剖析

——基于 0.12 内核

赵 炯 编著

机械工业出版社

本书对早期 Linux 内核（v0.12）全部代码文件进行了详细、全面的注释和说明，旨在帮助读者用较短的时间对 Linux 的工作机理获得全面而深刻的理解，为进一步学习和研究 Linux 打下坚实的基础。虽然选择的版本较低，但该内核已能够正常编译运行，并且其中已包括了 Linux 工作原理的精髓。书中首先以 Linux 源代码版本的变迁为主线，介绍了 Linux 的历史，同时着重说明了各个内核版本的主要区别和改进，给出了选择 0.12 版内核源代码作为研究对象的原因。在正式描述内核源代码之前，概要介绍了运行 Linux 的 PC 的硬件组成结构、编制内核使用的汇编语言和 C 语言扩展部分，并且重点说明了 80x86 处理器在保护模式下运行的编程方法。接着详细介绍了 Linux 内核源代码目录树组织结构，并依据该结构对所有内核程序和文件进行了注释和详细说明。有关代码注释的章节安排基本上都分为具体研究对象的概述、每个文件的功能介绍、代码内注释、代码中难点及相关资料介绍等部分。为了加深读者对内核工作原理的理解，书中最后一章给出了围绕 Linux 0.12 系统的多个实验，其中涉及的程序均可从本书配套网站（www.oldlinux.org）上下载。

本书适合 Linux 爱好者作为学习内核工作原理的自学参考书籍，也适合作为高校计算机专业学生学习操作系统课程的辅助和实践教材，还可供一般技术人员作为开发嵌入式系统的参考书使用。

图书在版编目（CIP）数据

Linux 内核完全剖析——基于 0.12 内核/ 赵炯编著. —北京：机械工业出版社，2009.1
ISBN 978-7-111-25047-0

Ⅰ. L… Ⅱ. 赵… Ⅲ. Linux 操作系统 Ⅳ. TP316.89

中国版本图书馆 CIP 数据核字（2008）第 134851 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：车 忱
责任编辑：车 忱
责任印制：李 妍
保定市中画美凯印刷有限公司印刷

2009 年 1 月 · 第 1 版第 1 次印刷
184mm×260mm · 60.25 印张 · 1494 千字
0001－4000 册
标准书号：ISBN 978-7-111-25047-0
定价：99.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话（010）68326294 68993821
购书热线电话（010）88379639 88379641 88379643
编辑热线电话（010）88379753 88379739

序

本书是一本有关 Linux 操作系统内核基本工作原理的入门读物。

本书的主要目标

本书的主要目标是使用尽量少的篇幅，对完整的 Linux 内核源代码进行解剖，使读者对操作系统的基本功能和实际实现方式获得全方位的理解。

本书读者应是知晓 Linux 系统的一般使用方法或具有一定的编程基础，但比较缺乏阅读目前最新内核源代码的基础知识，又急切希望能够进一步理解 UNIX 类操作系统内核工作原理和实际代码实现的爱好者。这部分读者的水平应该界于初级与中级水平之间。目前，这部分读者人数在 Linux 爱好者中所占的比例是很高的，而面向这部分读者以比较易懂和有效的手段讲解内核的书籍资料不多。

现有书籍不足之处

目前已有的描述 Linux 内核的书籍，均尽量选用最新 Linux 内核版本（例如 Fedora 8 使用的 2.6.24 稳定版等）进行描述，但由于目前 Linux 内核整个源代码的大小已经非常大（例如 2.2.20 版就已具有 268 万行代码！），因此这些书籍仅能对 Linux 内核源代码进行选择性或原理性的说明，许多系统实现细节被忽略。因此并不能使读者对实际 Linux 内核有清晰而完整的理解。

Scott Maxwell 的《Linux 内核源代码分析》基本上是面对 Linux 中、高级水平的读者，需要较为全面的基础知识才能完全理解。而且可能是由于篇幅所限，该书并没有对所有 Linux 内核代码进行注释，略去了很多内核实现细节，例如内核中使用的各个头文件(*.h)、生成内核代码映像文件的工具程序、各个 make 文件的作用和实现等均没有涉及。因此对于处于初、中级水平之间的读者来说阅读该书有些困难。

John Lions 的《莱昂氏 UNIX 源代码分析》虽然是一本学习 UNIX 类操作系统内核源代码很好的书，但是由于其采用的是 UNIX V6 版，其中系统调用等部分代码是用早已废弃的 PDP-11 系列机的汇编语言编制的，因此在阅读和理解与硬件部分相关的源代码时就会遇到较大的困难。

A. S. Tanenbaum 的《操作系统：设计与实现》是有关操作系统内核实现很好的入门书籍，但该书所叙述的 MINIX 系统是一种基于消息传递的内核实现机制，与 Linux 内核的实现有所区别。因此在学习该书之后，并不能很顺利地即刻着手进一步学习较新的 Linux 内核源代码实现。

在使用这些书籍进行学习时会有一种“盲人摸象”的感觉，不容易真正理解 Linux 内核系统具体实现的整体概念，尤其是对那些 Linux 系统初学者，或刚学会如何使用 Linux 系统的人在使用那些书学习内核原理时，内核的整体运作结构并不能清晰地在脑海中形成。这在本人多年的 Linux 内核学习过程中也深有体会。在 1991 年 10 月，Linux 的创始人 Linus Torvalds 在开发出 Linux 0.03 版后写的一篇文章中也提到了同样的问题。在这篇题为《Linux--a Free unix-386 Kernel》[⊖] 的文章中，他说：“开发 Linux 是为了那些操作系统爱好者和计算机科学系的学生使用、学习和娱乐”。“自由软件基金会的 GNU Hurd 系统如果开发出来就已经显得太庞大而不适合学习和理解。”而现今流行的 Linux 系统要比当年 GNU 的 Hurd 系统更为庞大和复杂，因此

⊖原文可参见：http://oldlinux.org/Linus/

同样也已经不适合作为操作系统初学者的入门学习起点。这也是作者基于 Linux 早期内核版本写作本书的动机之一。

阅读早期内核的其他好处

目前，已经出现不少基于 Linux 早期内核而开发的专门用于嵌入式系统的内核版本，如 DJJ 的 x86 操作系统、μClinux 等（在 www.linux.org 上有专门目录），世界上也有许多人认识到通过早期 Linux 内核源代码学习的好处，目前国内也已经有人正在组织人力注释出版类似本文的书籍。因此，通过阅读 Linux 早期内核版本的源代码，的确是学习 Linux 系统的一种行之有效的途径，并且对研究和应用 Linux 嵌入式系统也有很大的帮助。

在对早期内核源代码的注释过程中，作者发现，早期内核源代码几乎就是目前所使用的较新内核的一个精简版本。其中已经包括了目前新版本中几乎所有的基本功能原理的内容。正如《系统软件：系统编程导论》一书的作者 Leland L. Beck 在介绍系统程序以及操作系统设计时，引入了一种极其简化的简单指令计算机(SIC)系统来说明所有系统程序的设计和实现原理，从而既避免了实际计算机系统的复杂性，又能透彻地说明问题。这里选择 Linux 的早期内核版本作为学习对象，其指导思想与 Leland 是一致的。这对 Linux 内核学习的入门者来说，是最理想的选择之一。

对于那些已经比较熟悉内核工作原理的人，为了能让自己在实际工作中对系统的实际运转机制不产生一种空中楼阁的感觉，因此也有必要阅读内核源代码。

当然，使用早期内核作为学习的对象也有不足之处。所选用的 Linux 早期内核版本不支持虚拟文件系统（VFS）和网络系统也不包含对现有内核中复杂子系统的说明，而仅支持 a.out 执行文件。但由于本书是作为 Linux 内核工作机理实现的入门教材，因此这也正是选择早期内核版本的优点之一。通过学习本书，可以为进一步学习这些高级内容打下坚实的基础。

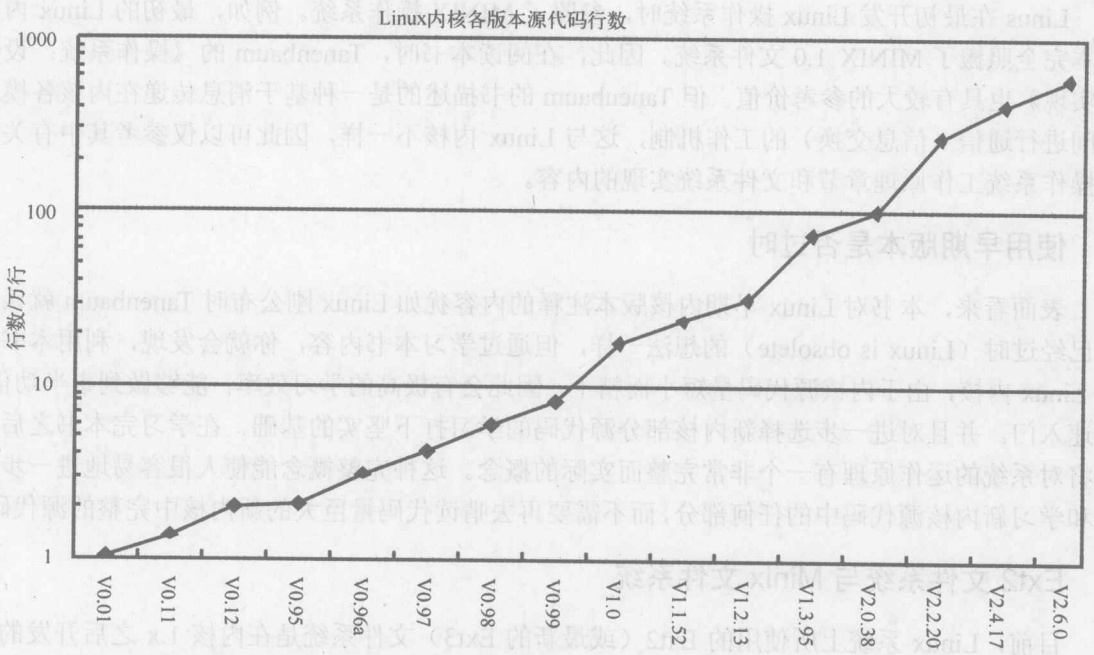
阅读完整源代码的重要性和必要性

正如 Linux 系统的创始人在一篇新闻组投稿上所说的，要理解一个软件系统的真正运行机制，一定要阅读其源代码。系统本身是一个完整的整体，具有很多看似不重要的细节，但是若忽略这些细节，就会对整个系统的理解带来困难，并且不能真正了解一个实际系统的实现方法和手段。

虽然阅读一些操作系统原理经典书籍如 M. J. Bach 的《UNIX 操作系统设计》，能够对 UNIX 类操作系统的工作原理有一些定了解，但实际上对操作系统的真正组成和内部关系实现的理解仍不是很清晰。正如 Tanenbaum 所说的，“许多操作系统教材都是重理论而轻实践”，“多数书籍和课程为调度算法耗费大量的时间和篇幅而完全忽略 I/O。其实，前者通常不足一页代码，而后者往往要占到整个系统三分之一的代码总量。”内核中大量的重要细节均未提到。因此并不能让读者理解一个真正的操作系统实现的奥妙所在。只有在详细阅读过完整的内核源代码之后，才会对系统有一种豁然开朗的感觉，对整个系统的运作过程有深刻的理解。以后再选择最新的或较新内核源代码进行学习时，也不会遇到大问题，基本上都能顺利地理解新代码的内容。

如何选择要阅读的内核代码版本

那么，如何选择既能达到上述要求，又不被太多的内容而搞乱头脑，学习效率又高呢？作者通过对大量内核版本进行比较和选择后，最终选择了与目前 Linux 内核基本功能较为相近，又非常短小的 0.12 版内核作为入门学习的最佳版本。下图是对一些主要 Linux 内核版本行数的统计。



目前的 Linux 内核源代码量都在几百万行的数量上，2.6.0 版内核代码行数约为 592 万行，极其庞大，对这些版本进行完全注释和说明几乎是不可能的。而 0.12 版内核不超过 2 万行代码量，因此完全可以在一本书中解释和注释清楚。麻雀虽小，五脏俱全。为了对所研究的系统有感性的了解，并能利用实验来加深对原理的理解，作者还专门重建了基于该内核的可运行的 Linux 0.12 系统。由于其中含有 GNU gcc 编译环境，因此使用该系统也能做一些简单的开发工作。

另外，使用该版本可以避免涉及较新内核版本中已经变得越来越复杂的各子系统（如 VFS、ext2 或 ext3 文件系统、网络子系统、新的复杂的内存管理机制等）。

阅读本书需具备的基础知识

在阅读本书时，读者必须具备一些基本的 C 语言知识和 Intel CPU 汇编语言知识。有关 C 语言最佳的参考资料仍然是 Brian W. Kernighan 和 Dennis M. Ritchie 编写的《The C Programming Language》一书。而汇编语言的资料则可以参考任意一本讲解与 Intel CPU 相关的汇编语言教材。另外，还需要一些嵌入式汇编语言的资料。有关嵌入式汇编的权威信息都包含在 GNU gcc 编译器手册中。我们也可以从 Internet 上搜索到一些有关嵌入式汇编的比较有价值的短文。本书中也包含了一些关于嵌入式汇编的基本语法说明。

除此之外，还希望读者具备以下一些基础知识或者有相关的参考书籍在身边。其一是有关 80x86 处理器结构和编程的知识或资料。例如，可以从网上下载的 80x86 编程手册（80386 Programmer's Reference Manual）；其二是有关 80x86 硬件体系结构和接口编程的知识或资料（有关这方面的资料很多）；其三还应具备使用 Linux 系统的简单技能。

另外，由于 Linux 系统内核的实现最早是根据 M. J. Bach 的《UNIX 操作系统设计》一书的基本原理开发的，源代码中许多变量或函数的名称都来自该书，因此在阅读本书时若能适当参考该书，会更易于理解内核源代码。

Linus 在最初开发 Linux 操作系统时，参照了 MINIX 操作系统。例如，最初的 Linux 内核版本完全照搬了 MINIX 1.0 文件系统。因此，在阅读本书时，Tanenbaum 的《操作系统：设计与实现》也具有较大的参考价值。但 Tanenbaum 的书描述的是一种基于消息传递在内核各模块之间进行通信（信息交换）的工作机制，这与 Linux 内核不一样，因此可以仅参考其中有关一般操作系统工作原理章节和文件系统实现的内容。

使用早期版本是否过时

表面看来，本书对 Linux 早期内核版本注释的内容犹如 Linux 刚公布时 Tanenbaum 就认为其已经过时（Linux is obsolete）的想法一样，但通过学习本书内容，你就会发现，利用本书学习 Linux 内核，由于内核源代码量短小而精干，因此会有极高的学习效率，能够做到事半功倍，快速入门。并且对进一步选择新内核部分源代码的学习打下坚实的基础。在学习完本书之后，你将对系统的运作原理有一个非常完整而实际的概念。这种完整概念能使人很容易地进一步选择和学习新内核源代码中的任何部分，而不需要再去啃读代码量巨大的新内核中完整的源代码。

Ext2 文件系统与 Minix 文件系统

目前，Linux 系统上所使用的 Ext2（或最新的 Ext3）文件系统是在内核 1.x 之后开发的。其功能详尽并且性能也非常稳定，是目前 Linux 操作系统上默认的标准文件系统。但是，作为对 Linux 操作系统完整工作原理入门学习所使用的部分，原则上是越精简越好。为了对一个操作系统有完整的理解，并且不受其中各子系统中复杂和过多的细节的干扰，在选择学习剖析用的内核版本时，只要系统的部分代码内容能说明实际工作原理，就越简单越好。

Linux 内核 0.12 版仅包含最为简单的 MINIX 1.0 文件系统，对于理解一个操作系统中文件系统的实际组成和工作原理已经足够。这也是选择 Linux 早期内核版本进行学习的主要原因之一。

在完整阅读本书之后，相信您定会发出这样的感叹：对于 Linux 内核系统，我现在终于入门了！此时，您应该有十分的把握去进一步学习最新 Linux 内核中各部分的工作原理和过程了。

本书与《Linux 内核完全剖析》的主要区别

本书采用了 Linux 0.12 版内核作为学习和剖析对象，而作者的《Linux 内核完全剖析》（机械工业出版社，2006）则是围绕着 Linux 0.11 内核版本进行描述的。Linux 0.12 版内核发布于 1992 年年初，它改正了 0.11 版中存在的一些错误，并提供了一些新的功能。Linux 0.12 版内核增加的新功能主要有：符号链接、虚拟终端支持、select()函数支持和数学协处理器模拟代码实现。由于该版本内核在稳定性方面的出色表现，Linus 曾提及 0.12 版内核是他在内核开发过程中最为满意的版本之一。

本书包含《Linux 内核完全剖析》的所有内容，并在此基础上添加了一些与 0.12 版内核新功能相关的硬件信息。例如本书提供了比较完整的有关数学协处理器的资料。在写作布局和风格上本书保持着与《Linux 内核完全剖析》相同的风格。书中章节设置顺序和注释风格没有变化。

当然，本书也改正了《Linux 内核完全剖析》中的一些明显错误。这要感谢读者对本书提供的反馈以及热情的支持。

同济大学
赵炯 博士

目　录

序

第 1 章　概述 …… 1

1.1　Linux 的诞生和发展 …… 1

1.1.1　UNIX 操作系统的诞生 …… 1

1.1.2　MINIX 操作系统 …… 1

1.1.3　GNU 计划 …… 2

1.1.4　POSIX 标准 …… 2

1.1.5　Linux 操作系统的诞生 …… 3

1.1.6　Linux 操作系统版本的变迁 …… 4

1.1.7　Linux 名称的由来 …… 6

1.1.8　早期 Linux 系统开发的主要贡献者 … 7

1.2　内容综述 …… 8

1.3　本章小结 …… 12

第 2 章　微型计算机组成结构 …… 13

2.1　微型计算机组成原理 …… 13

2.2　I/O 端口寻址和访问控制方式 …… 15

2.2.1　I/O 端口和寻址 …… 15

2.2.2　接口访问控制 …… 17

2.3　主存储器、BIOS 和 CMOS 存储器 …… 17

2.3.1　主存储器 …… 17

2.3.2　基本输入/输出程序 BIOS …… 18

2.3.3　CMOS 存储器 …… 19

2.4　控制器和控制卡 …… 19

2.4.1　中断控制器 …… 19

2.4.2　DMA 控制器 …… 20

2.4.3　定时/计数器 …… 21

2.4.4　键盘控制器 …… 21

2.4.5　串行控制卡 …… 22

2.4.6　显示控制 …… 24

2.4.7　软盘和硬盘控制器 …… 25

2.5　本章小结 …… 28

第 3 章　内核编程语言和环境 …… 29

3.1　as86 汇编器 …… 29

3.1.1　as86 汇编语言语法 …… 30

3.1.2　as86 汇编语言程序 …… 31

3.1.3　as86 汇编语言程序的编译和链接 … 33

3.1.4　as86 和 ld86 使用方法和选项 …… 34

3.2　GNU as 汇编 …… 35

3.2.1　编译 as 汇编语言程序 …… 36

3.2.2　as 汇编语法 …… 37

3.2.3　指令语句、操作数和寻址 …… 38

3.2.4　区与重定位 …… 41

3.2.5　符号 …… 43

3.2.6　as 汇编命令 …… 44

3.2.7　编写 16 位代码 …… 46

3.2.8　AS 汇编器命令行选项 …… 46

3.3　C 语言程序 …… 46

3.3.1　C 程序编译和链接 …… 46

3.3.2　嵌入汇编 …… 47

3.3.3　圆括号中的组合语句 …… 51

3.3.4　寄存器变量 …… 52

3.3.5　内联函数 …… 52

3.4　C 与汇编程序的相互调用 …… 54

3.4.1　C 函数调用机制 …… 54

3.4.2　在汇编程序中调用 C 函数 …… 59

3.4.3　在 C 程序中调用汇编函数 …… 61

3.5　Linux 0.12 目标文件格式 …… 63

3.5.1　目标文件格式 …… 63

3.5.2　Linux 0.12 中的目标文件格式 …… 66

3.5.3　链接程序输出 …… 68

3.5.4　链接程序预定义变量 …… 69

3.5.5　System.map 文件 …… 70

3.6　Make 程序和 Makefile 文件 …… 72

3.6.1　Makefile 文件内容 …… 72

3.6.2　Makefile 文件中的规则 …… 73

3.6.3　Makefile 文件示例 …… 73

3.6.4　make 处理 Makefile 文件的方式 … 75

3.6.5　Makefile 中的变量 …… 76

3.6.6 让 make 自动推断命令 …… 76
3.6.7 隐含规则中的自动变量 …… 77
3.7 本章小结 …… 78
第 4 章 80x86 保护模式及其编程 …… 79
4.1 80x86 系统寄存器和系统指令 …… 79
4.1.1 标志寄存器 …… 79
4.1.2 内存管理寄存器 …… 80
4.1.3 控制寄存器 …… 81
4.1.4 系统指令 …… 84
4.2 保护模式内存管理 …… 85
4.2.1 内存寻址 …… 85
4.2.2 地址变换 …… 86
4.2.3 保护 …… 88
4.3 分段机制 …… 89
4.3.1 段的定义 …… 89
4.3.2 段描述符表 …… 91
4.3.3 段选择符 …… 93
4.3.4 段描述符 …… 95
4.3.5 代码和数据段描述符类型 …… 98
4.3.6 系统描述符类型 …… 99
4.4 分页机制 …… 100
4.4.1 页表结构 …… 102
4.4.2 页表项格式 …… 103
4.4.3 虚拟存储 …… 104
4.5 保护 …… 104
4.5.1 段级保护 …… 105
4.5.2 访问数据段时的特权级检查 …… 107
4.5.3 代码段之间转移控制时的特权级检查 …… 108
4.5.4 页级保护 …… 113
4.5.5 组合页级和段级保护 …… 115
4.6 中断和异常处理 …… 115
4.6.1 异常和中断向量 …… 115
4.6.2 中断源和异常源 …… 116
4.6.3 异常分类 …… 117
4.6.4 程序或任务的重新执行 …… 117
4.6.5 开启和禁止中断 …… 118
4.6.6 异常和中断的优先级 …… 118
4.6.7 中断描述符表 …… 119
4.6.8 IDT 描述符 …… 119
4.6.9 异常与中断处理 …… 120
4.6.10 中断处理任务 …… 123
4.6.11 错误码 …… 123
4.7 任务管理 …… 124
4.7.1 任务的结构和状态 …… 125
4.7.2 任务的执行 …… 126
4.7.3 任务管理数据结构 …… 126
4.7.4 任务切换 …… 129
4.7.5 任务链 …… 131
4.7.6 任务地址空间 …… 132
4.8 保护模式编程初始化 …… 133
4.8.1 进入保护模式时的初始化操作 …… 134
4.8.2 模式切换 …… 135
4.9 一个简单的多任务内核实例 …… 137
4.9.1 多任务程序结构和工作原理 …… 137
4.9.2 引导启动程序 boot.s …… 140
4.9.3 多任务内核程序 head.s …… 142
第 5 章 Linux 内核体系结构 …… 147
5.1 Linux 内核模式 …… 147
5.2 Linux 内核系统体系结构 …… 148
5.3 Linux 内核对内存的管理和使用 …… 150
5.3.1 物理内存 …… 150
5.3.2 内存地址空间概念 …… 150
5.3.3 内存分段机制 …… 151
5.3.4 内存分页管理 …… 154
5.3.5 CPU 多任务和保护方式 …… 157
5.3.6 虚拟地址、线性地址和物理地址之间的关系 …… 157
5.3.7 用户申请内存的动态分配 …… 161
5.4 中断机制 …… 162
5.4.1 中断操作原理 …… 162
5.4.2 80x86 微机的中断子系统 …… 163
5.4.3 中断向量表 …… 164
5.4.4 Linux 内核的中断处理 …… 164
5.4.5 标志寄存器的中断标志 …… 166
5.5 Linux 的系统调用 …… 166
5.5.1 系统调用接口 …… 166
5.5.2 系统调用处理过程 …… 167

5.5.3 Linux 系统调用的参数传递方式 …… 168
5.6 系统时间和定时 …… 168
5.6.1 系统时间 …… 168
5.6.2 系统定时 …… 169
5.7 Linux 进程控制 …… 170
5.7.1 任务数据结构 …… 170
5.7.2 进程运行状态 …… 175
5.7.3 进程初始化 …… 176
5.7.4 创建新进程 …… 177
5.7.5 进程调度 …… 178
5.7.6 终止进程 …… 179
5.8 Linux 系统中堆栈的使用方法 …… 179
5.8.1 初始化阶段 …… 180
5.8.2 任务的堆栈 …… 181
5.8.3 任务内核态堆栈与用户态堆栈之间的切换 …… 183
5.9 Linux 0.12 采用的文件系统 …… 184
5.10 Linux 内核源代码的目录结构 …… 184
5.10.1 内核主目录 linux …… 185
5.10.2 引导启动程序目录 boot …… 185
5.10.3 文件系统目录 fs …… 186
5.10.4 头文件主目录 include …… 187
5.10.5 内核初始化程序目录 init …… 188
5.10.6 内核程序主目录 kernel …… 188
5.10.7 内核库函数目录 lib …… 191
5.10.8 内存管理程序目录 mm …… 191
5.10.9 编译内核工具程序目录 tools … 192
5.11 内核系统与应用程序的关系 …… 192
5.12 linux/Makefile 文件 …… 192
5.12.1 功能描述 …… 193
5.12.2 代码注释 …… 194
5.13 本章小结 …… 198
第 6 章 引导启动程序 …… 199
6.1 总体功能 …… 199
6.2 bootsect.S 程序 …… 201
6.2.1 功能描述 …… 201
6.2.2 代码注释 …… 202
6.2.3 其他信息 …… 213
6.3 setup.S 程序 …… 214
6.3.1 功能描述 …… 214
6.3.2 代码注释 …… 216
6.3.3 其他信息 …… 232
6.4 head.s 程序 …… 242
6.4.1 功能描述 …… 242
6.4.2 代码注释 …… 243
6.4.3 其他信息 …… 251
6.5 本章小结 …… 253
第 7 章 初始化程序 …… 255
7.1 main.c 程序 …… 255
7.1.1 功能描述 …… 255
7.1.2 代码注释 …… 258
7.1.3 其他信息 …… 265
7.2 环境初始化工作 …… 268
7.3 本章小结 …… 269
第 8 章 内核代码 …… 271
8.1 总体功能 …… 271
8.1.1 中断处理程序 …… 271
8.1.2 系统调用处理相关程序 …… 272
8.1.3 其他通用类程序 …… 273
8.2 asm.s 程序 …… 273
8.2.1 功能描述 …… 273
8.2.2 代码注释 …… 275
8.2.3 Intel 保留中断向量的定义 …… 279
8.3 traps.c 程序 …… 279
8.3.1 功能描述 …… 279
8.3.2 代码注释 …… 279
8.4 sys_call.s 程序 …… 284
8.4.1 功能描述 …… 284
8.4.2 代码注释 …… 286
8.4.3 其他信息 …… 294
8.5 mktime.c 程序 …… 296
8.5.1 功能描述 …… 296
8.5.2 代码注释 …… 297
8.5.3 闰年的计算方法 …… 298
8.6 sched.c 程序 …… 298
8.6.1 功能描述 …… 298

8.6.2 代码注释 …… 300
8.6.3 其他信息 …… 312
8.7 signal.c 程序 …… 317
8.7.1 功能描述 …… 317
8.7.2 代码注释 …… 325
8.7.3 进程信号说明 …… 331
8.8 exit.c 程序 …… 332
8.8.1 功能描述 …… 332
8.8.2 代码注释 …… 333
8.9 fork.c 程序 …… 345
8.9.1 功能描述 …… 345
8.9.2 代码注释 …… 346
8.9.3 任务状态段信息 …… 351
8.10 sys.c 程序 …… 352
8.10.1 功能描述 …… 352
8.10.2 代码注释 …… 353
8.11 vsprintf.c 程序 …… 366
8.11.1 功能描述 …… 366
8.11.2 代码注释 …… 367
8.11.3 vsprintf()的格式字符串 …… 372
8.11.4 与当前版本的区别 …… 374
8.12 printk.c 程序 …… 374
8.12.1 功能描述 …… 374
8.12.2 代码注释 …… 375
8.13 panic.c 程序 …… 375
8.13.1 功能描述 …… 375
8.13.2 代码注释 …… 376
8.14 本章小结 …… 376
第 9 章 块设备驱动程序 …… 377
9.1 总体功能 …… 378
9.1.1 块设备请求项和请求队列 …… 378
9.1.2 块设备访问调度处理 …… 380
9.1.3 块设备操作方式 …… 380
9.2 blk.h 文件 …… 381
9.2.1 功能描述 …… 381
9.2.2 代码注释 …… 382
9.3 hd.c 程序 …… 386
9.3.1 功能描述 …… 386
9.3.2 代码注释 …… 388
9.3.3 其他信息 …… 400
9.4 ll_rw_blk.c 程序 …… 409
9.4.1 功能描述 …… 409
9.4.2 代码注释 …… 409
9.5 ramdisk.c 程序 …… 415
9.5.1 功能描述 …… 415
9.5.2 代码注释 …… 417
9.6 floppy.c 程序 …… 421
9.6.1 功能描述 …… 421
9.6.2 代码注释 …… 422
9.6.3 其他信息 …… 435
第 10 章 字符设备驱动程序 …… 449
10.1 总体功能 …… 449
10.1.1 终端驱动程序基本原理 …… 449
10.1.2 Linux 支持的终端设备类型 …… 450
10.1.3 终端基本数据结构 …… 451
10.1.4 规范模式和非规范模式 …… 455
10.1.5 控制台终端和串行终端设备 …… 456
10.1.6 终端驱动程序接口 …… 459
10.2 keyboard.S 程序 …… 459
10.2.1 功能描述 …… 459
10.2.2 代码注释 …… 459
10.2.3 其他信息 …… 473
10.3 console.c 程序 …… 477
10.3.1 功能描述 …… 477
10.3.2 代码注释 …… 477
10.3.3 其他信息 …… 503
10.4 serial.c 程序 …… 510
10.4.1 功能描述 …… 510
10.4.2 代码注释 …… 510
10.4.3 异步串行通信控制器 UART …… 512
10.5 rs_io.s 程序 …… 517
10.5.1 功能描述 …… 517
10.5.2 代码注释 …… 518
10.6 tty_io.c 程序 …… 522
10.6.1 功能描述 …… 522
10.6.2 代码注释 …… 523
10.6.3 控制字符 VTIME、VMIN …… 536
10.7 tty_ioctl.c 程序 …… 536
10.7.1 功能描述 …… 536
10.7.2 代码注释 …… 537

10.7.3 波特率与波特率因子 …… 543
第 11 章 数学协处理器 …… 545
11.1 总体功能描述 …… 545
11.1.1 浮点数据类型 …… 546
11.1.2 数学协处理器功能和结构 …… 550
11.2 math_emulate.c 程序 …… 553
11.2.1 功能描述 …… 553
11.2.2 代码注释 …… 555
11.3 error.c 程序 …… 565
11.3.1 功能描述 …… 565
11.3.2 代码注释 …… 566
11.4 ea.c 程序 …… 566
11.4.1 功能描述 …… 566
11.4.2 代码注释 …… 567
11.5 convert.c 程序 …… 570
11.5.1 功能描述 …… 570
11.5.2 代码注释 …… 570
11.6 add.c 程序 …… 574
11.6.1 功能描述 …… 574
11.6.2 代码注释 …… 575
11.7 compare.c 程序 …… 577
11.7.1 功能描述 …… 577
11.7.2 代码注释 …… 577
11.8 get_put.c 程序 …… 579
11.8.1 功能描述 …… 579
11.8.2 代码注释 …… 579
11.9 mul.c 程序 …… 585
11.9.1 功能描述 …… 585
11.9.2 代码注释 …… 585
11.10 div.c 程序 …… 586
11.10.1 功能描述 …… 586
11.10.2 代码注释 …… 587
第 12 章 文件系统 …… 589
12.1 总体功能 …… 589
12.1.1 MINIX 文件系统 …… 590
12.1.2 文件类型、属性和目录项 …… 594
12.1.3 高速缓冲区 …… 598
12.1.4 文件系统底层函数 …… 599
12.1.5 文件中数据的访问操作 …… 599
12.1.6 文件和目录管理系统调用 …… 601
12.1.7 360KB 软盘中文件系统实例分析 …… 602
12.2 buffer.c 程序 …… 605
12.2.1 功能描述 …… 605
12.2.2 代码注释 …… 611
12.3 bitmap.c 程序 …… 621
12.3.1 功能描述 …… 622
12.3.2 代码注释 …… 622
12.4 truncate.c 程序 …… 627
12.4.1 功能描述 …… 627
12.4.2 代码注释 …… 627
12.5 inode.c 程序 …… 630
12.5.1 功能描述 …… 630
12.5.2 代码注释 …… 632
12.6 super.c 程序 …… 641
12.6.1 功能描述 …… 641
12.6.2 代码注释 …… 642
12.7 namei.c 程序 …… 650
12.7.1 功能描述 …… 650
12.7.2 代码注释 …… 651
12.8 file_table.c 程序 …… 675
12.8.1 功能描述 …… 675
12.8.2 代码注释 …… 675
12.9 block_dev.c 程序 …… 676
12.9.1 功能描述 …… 676
12.9.2 代码注释 …… 677
12.10 file_dev.c 程序 …… 680
12.10.1 功能描述 …… 680
12.10.2 代码注释 …… 680
12.11 pipe.c 程序 …… 682
12.11.1 功能描述 …… 682
12.11.2 代码注释 …… 683
12.12 char_dev.c 程序 …… 687
12.12.1 功能描述 …… 687
12.12.2 代码注释 …… 687
12.13 read_write.c 程序 …… 690
12.13.1 功能描述 …… 690
12.13.2 代码注释 …… 690
12.13.3 用户程序读写操作过程 …… 693
12.14 open.c 程序 …… 695

12.14.1 功能描述 …… 696
12.14.2 代码注释 …… 696
12.15 exec.c 程序 …… 703
12.15.1 功能描述 …… 703
12.15.2 代码注释 …… 706
12.15.3 其他信息 …… 718
12.16 stat.c 程序 …… 722
12.16.1 功能描述 …… 722
12.16.2 代码注释 …… 722
12.17 fcntl.c 程序 …… 724
12.17.1 功能描述 …… 724
12.17.2 代码注释 …… 725
12.18 ioctl.c 程序 …… 727
12.18.1 功能描述 …… 727
12.18.2 代码注释 …… 728
12.19 select.c 程序 …… 729
12.19.1 功能描述 …… 729
12.19.2 代码注释 …… 733
第 13 章 内存管理 …… 741
13.1 总体功能 …… 741
13.1.1 内存分页管理机制 …… 741
13.1.2 Linux 中物理内存的管理和分配 …… 744
13.1.3 Linux 内核对线性地址空间的使用分配 …… 745
13.1.4 页面出错异常处理 …… 745
13.1.5 写时复制机制 …… 746
13.1.6 需求加载机制 …… 746
13.2 memory.c 程序 …… 747
13.2.1 功能描述 …… 747
13.2.2 代码注释 …… 749
13.3 page.s 程序 …… 765
13.3.1 功能描述 …… 765
13.3.2 代码注释 …… 765
13.3.3 页出错异常处理 …… 766
13.4 swap.c 程序 …… 767
13.4.1 功能描述 …… 767
13.4.2 代码注释 …… 767
第 14 章 头文件 …… 775
14.1 include/目录下的文件 …… 775
14.2 a.out.h 文件 …… 776
14.2.1 功能描述 …… 776
14.2.2 代码注释 …… 777
14.2.3 a.out 执行文件格式 …… 783
14.3 const.h 文件 …… 786
14.3.1 功能描述 …… 786
14.3.2 代码注释 …… 786
14.4 ctype.h 文件 …… 787
14.4.1 功能描述 …… 787
14.4.2 代码注释 …… 787
14.5 errno.h 文件 …… 788
14.5.1 功能描述 …… 788
14.5.2 代码注释 …… 789
14.6 fcntl.h 文件 …… 790
14.6.1 功能描述 …… 790
14.6.2 代码注释 …… 790
14.7 signal.h 文件 …… 792
14.7.1 功能描述 …… 792
14.7.2 文件注释 …… 792
14.8 stdarg.h 文件 …… 795
14.8.1 功能描述 …… 795
14.8.2 代码注释 …… 795
14.9 stddef.h 文件 …… 796
14.9.1 功能描述 …… 796
14.9.2 代码注释 …… 796
14.10 string.h 文件 …… 797
14.10.1 功能描述 …… 797
14.10.2 代码注释 …… 797
14.11 termios.h 文件 …… 806
14.11.1 功能描述 …… 806
14.11.2 代码注释 …… 807
14.11.3 控制字符 TIME 和 MIN …… 812
14.12 time.h 文件 …… 813
14.12.1 功能描述 …… 813
14.12.2 代码注释 …… 813
14.13 unistd.h 文件 …… 815
14.13.1 功能描述 …… 815
14.13.2 代码注释 …… 815
14.14 utime.h 文件 …… 821
14.14.1 功能描述 …… 821

14.14.2 代码注释 …… 821
14.15 include/asm/目录下的文件 … 821
14.16 io.h 文件 …… 821
14.16.1 功能描述 …… 821
14.16.2 代码注释 …… 822
14.17 memory.h 文件 …… 822
14.17.1 功能描述 …… 822
14.17.2 代码注释 …… 822
14.18 segment.h 文件 …… 823
14.18.1 功能描述 …… 823
14.18.2 代码注释 …… 823
14.19 system.h 文件 …… 825
14.19.1 功能描述 …… 825
14.19.2 代码注释 …… 827
14.20 include/linux/目录下的文件 … 829
14.21 config.h 文件 …… 829
14.21.1 功能描述 …… 829
14.21.2 代码注释 …… 829
14.22 fdreg.h 头文件 …… 831
14.22.1 功能描述 …… 831
14.22.2 文件注释 …… 832
14.23 fs.h 文件 …… 834
14.23.1 功能描述 …… 834
14.23.2 代码注释 …… 834
14.24 hdreg.h 文件 …… 839
14.24.1 功能描述 …… 839
14.24.2 代码注释 …… 840
14.24.3 硬盘分区表 …… 841
14.25 head.h 文件 …… 842
14.25.1 功能描述 …… 842
14.25.2 代码注释 …… 842
14.26 kernel.h 文件 …… 843
14.26.1 功能描述 …… 843
14.26.2 代码注释 …… 843
14.27 mm.h 文件 …… 844
14.27.1 功能描述 …… 844
14.27.2 代码注释 …… 844
14.28 sched.h 文件 …… 845
14.28.1 功能描述 …… 845
14.28.2 代码注释 …… 846
14.29 sys.h 文件 …… 853
14.29.1 功能描述 …… 853
14.29.2 代码注释 …… 854
14.30 tty.h 文件 …… 856
14.30.1 功能描述 …… 856
14.30.2 代码注释 …… 856
14.31 include/sys/目录中的文件 …… 859
14.32 param.h 文件 …… 859
14.32.1 功能描述 …… 859
14.32.2 代码注释 …… 859
14.33 resource.h 文件 …… 859
14.33.1 功能描述 …… 859
14.33.2 代码注释 …… 860
14.34 stat.h 文件 …… 862
14.34.1 功能描述 …… 862
14.34.2 代码注释 …… 862
14.35 time.h 文件 …… 863
14.35.1 功能描述 …… 863
14.35.2 代码注释 …… 863
14.36 times.h 文件 …… 865
14.36.1 功能描述 …… 865
14.36.2 代码注释 …… 865
14.37 types.h 文件 …… 865
14.37.1 功能描述 …… 865
14.37.2 代码注释 …… 865
14.38 utsname.h 文件 …… 866
14.38.1 功能描述 …… 866
14.38.2 代码注释 …… 867
14.39 wait.h 文件 …… 867
14.39.1 功能描述 …… 867
14.39.2 代码注释 …… 867
第 15 章 库文件 …… 869
15.1 _exit.c 程序 …… 869
15.1.1 功能描述 …… 869
15.1.2 代码注释 …… 870
15.1.3 相关信息 …… 870
15.2 close.c 程序 …… 870
15.2.1 功能描述 …… 870
15.2.2 代码注释 …… 870
15.3 ctype.c 程序 …… 871

15.3.1 功能描述 …… 871
15.3.2 代码注释 …… 871
15.4 dup.c 程序 …… 872
15.4.1 功能描述 …… 872
15.4.2 代码注释 …… 872
15.5 errno.c 程序 …… 872
15.5.1 功能描述 …… 872
15.5.2 代码注释 …… 872
15.6 execve.c 程序 …… 873
15.6.1 功能描述 …… 873
15.6.2 代码注释 …… 873
15.7 malloc.c 程序 …… 873
15.7.1 功能描述 …… 873
15.7.2 代码注释 …… 875
15.8 open.c 程序 …… 882
15.8.1 功能描述 …… 882
15.8.2 代码注释 …… 882
15.9 setsid.c 程序 …… 883
15.9.1 功能描述 …… 883
15.9.2 代码注释 …… 883
15.10 string.c 程序 …… 884
15.10.1 功能描述 …… 884
15.10.2 代码注释 …… 884
15.11 wait.c 程序 …… 884
15.11.1 功能描述 …… 884
15.11.2 代码注释 …… 885
15.12 write.c 程序 …… 885
15.12.1 功能描述 …… 885
15.12.2 代码注释 …… 885
第 16 章 建造工具 …… 887
16.1 build.c 程序 …… 887
16.1.1 功能描述 …… 887
16.1.2 代码注释 …… 888
16.2 MINIX 可执行文件头部数据结构 …… 893
第 17 章 实验环境设置与使用方法 …… 895
17.1 Bochs 仿真软件系统 …… 895
17.1.1 设置 Bochs 系统 …… 896
17.1.2 配置文件 *.bxrc …… 897
17.2 在 Bochs 中运行 Linux 0.1x 系统 …… 900
17.2.1 软件包中文件说明 …… 900
17.2.2 安装 Bochs 模拟系统 …… 902
17.2.3 运行 Linux 0.1x 系统 …… 902
17.3 访问磁盘映像文件中的信息 …… 904
17.3.1 使用 WinImage 工具软件 …… 904
17.3.2 利用现有 Linux 系统 …… 905
17.4 编译运行简单内核示例程序 …… 906
17.5 利用 Bochs 调试内核 …… 908
17.5.1 运行 Bochs 调试程序 …… 908
17.5.2 定位内核中的变量或数据结构 …… 914
17.6 创建磁盘映像文件 …… 915
17.6.1 利用 Bochs 软件自带的 Image 生成工具 …… 916
17.6.2 在 Linux 系统下使用 dd 命令创建 Image 文件 …… 917
17.6.3 利用 WinImage 创建 DOS 格式的软盘 Image 文件 …… 917
17.7 制作根文件系统 …… 918
17.7.1 根文件系统和根文件设备 …… 918
17.7.2 创建文件系统 …… 919
17.7.3 Linux-0.12 的 Bochs 配置文件 …… 921
17.7.4 在 hdc.img 上建立根文件系统 …… 922
17.7.5 使用硬盘 Image 上的根文件系统 …… 924
17.8 在 Linux 0.12 系统中编译 0.12 内核 …… 925
17.9 在 Fedora 系统中编译 Linux 0.1x 内核 …… 926
17.9.1 修改 Makefile 文件 …… 926
17.9.2 修改汇编程序中的注释 …… 927
17.9.3 内存位置对齐语句 align 值的修改 …… 927
17.9.4 修改嵌入宏汇编程序 …… 927
17.9.5 C 程序变量在汇编语句中的引用表示 …… 928
17.9.6 保护模式下调试显示函数 …… 928
17.10 内核引导启动+根文件系统组成的集成盘 …… 929

17.10.1 集成盘制作原理 …… 929
17.10.2 集成盘的制作过程 …… 931
17.10.3 运行集成盘系统 …… 933
17.11 利用 GDB 和 Bochs 调试内核源代码 …… 933
17.11.1 编译带 gdbstub 的 Bochs 系统 933
17.11.2 编译带调试信息的 Linux 0.1x 内核 …… 934
17.11.3 调试方法和步骤 …… 935
附录 …… 939
附录 A ASCII 码表 …… 939
附录 B 常用 C0、C1 控制字符表 … 940
附录 C 常用转义序列和控制序列 …… 941
附录 D 第 1 套键盘扫描码集 …… 943
参考文献 …… 944

第1章 概 述

本章首先回顾了 Linux 操作系统的诞生、开发和成长过程，由此读者可以理解本书选择 Linux 系统早期版本作为学习对象的一些原因；然后具体说明了选择早期 Linux 内核版本进行学习的优点和不足之处以及如何开始进一步学习；最后对各章的内容进行了简要介绍。

1.1 Linux 的诞生和发展

Linux 操作系统是 UNIX 操作系统的一种克隆系统。它诞生于 1991 年 10 月 5 日（这是第一次正式向外公布的时间）。此后借助于 Internet 网络，经过全世界计算机爱好者的共同努力，现已成为当今使用最多的一种 UNIX 类操作系统，并且使用人数还在迅猛增长。

Linux 操作系统的诞生、发展和成长过程依赖于以下五个重要支柱：UNIX 操作系统、MINIX 操作系统、GNU 计划、POSIX 标准和 Internet。下面根据这五个基本线索来回顾一下 Linux 的酝酿过程、开发历程以及最初的发展。首先分别介绍其中的四个基本要素，然后根据 Linux 的创始人 Linus Torvalds 从对计算机感兴趣而自学计算机知识，到心里开始酝酿编制一个自己的操作系统，到最初 Linux 内核 0.01 版公布以及从此如何艰难地一步一个脚印地在全世界黑客的帮助下推出比较完善的 1.0 版本这段经过，对 Linux 的早期发展历史进行详细介绍。

当然，目前 Linux 内核版本已经开发到了 2.6.x 版。而大多数 Linux 系统中所用到的内核是稳定的 2.6.12 版内核（其中第 2 个数字若是奇数则表示正在开发的版本，不能保证系统的稳定性）。对于 Linux 的一般发展史，许多文章和书籍都有介绍，这里不再重复。

1.1.1 UNIX 操作系统的诞生

UNIX 操作系统最早是美国贝尔实验室的 Ken Thompson 于 1969 年夏在 DEC PDP-7 小型计算机上开发的一个分时操作系统。

Ken Thompson 为了能在闲置不用的 PDP-7 计算机上运行他非常喜欢的星际旅行（Star Trek）游戏，于 1969 年夏天趁他夫人回家乡加利福尼亚度假期间，在一个月内开发出了 UNIX 操作系统的原型。当时使用的是 BCPL 语言（基本组合编程语言），后经 Dennis Ritchie 于 1972 年用移植性很强的 C 语言进行了改写，使得 UNIX 系统在大学得到了推广。

1.1.2 MINIX 操作系统

MINIX 系统是由 Andrew S. Tanenbaum 开发的。Tanenbaum 在荷兰阿姆斯特丹的 Vrije 大学数学与计算机科学系工作，是 ACM 和 IEEE 的资深会员（全世界也只有很少人是两会的资深会员），发表了 100 多篇文章，编写了 5 本计算机书籍。

Tanenbaum 虽出生在美国纽约，却是荷兰侨民（1914 年他的祖辈来到美国）。他在纽约上中学，在 M. I. T 上大学，在加州大学伯克利分校念博士学位。由于读博士后的缘故，他来到了家乡荷兰，从此就与家乡一直有来往，后来就在 Vrije 大学教书、带研究生。阿姆斯特丹是个常年阴雨绵绵的城市，但对于 Tanenbaum 来说，这最好不过了，因为在这样的环境下他就可以经