

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，由此可以理解本书为什么会选择 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的学习。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 年的 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，经过全世界各地计算机爱好者的共同努力，现已成为当今世界上使用最多的一种 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 网络。下面根据这五个基本线索来追寻一下 Linux 的开发历程、它的酝酿过程以及最初的发展经历。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 hacker 的帮助下最后推出比较完善的 1.0 版本这段时间的发展经过，也即对 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 2.5.52 版。而大多数 Linux 系统中所用到的内核是稳定的 2.4.20 版内核（其中第 2 个数字若是奇数则表示是正在开发的版本，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，这里就不重复。

1.1.1 UNIX 操作系统的诞生

Linux 操作系统是 UNIX 操作系统的一个克隆版本。UNIX 操作系统是美国贝尔实验室的 Ken.Thompson 和 Dennis Ritchie 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（Space travel）游戏，于是在 1969 年夏天乘他夫人回家乡加利福尼亚渡假期间，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 系统在大专院校得到了推广。

1.1.2 MINIX 操作系统

MINIX 系统是由 Andrew S. Tanenbaum（AST）开发的。AST 是在荷兰 Amsterdam 的 Vrije 大学数学与计算机科学系统工作，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）。共发表了 100 多篇文章，5 本计算机书籍。

AST 虽出生在美国纽约，但却是荷兰侨民（1914 年他的祖辈来到美国）。他在纽约上的中学、M.I.T 上的大学、加州大学 Berkeley 分校念的博士学位。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 大学开始教书、带研究生。荷兰首都 Amsterdam 是个常年阴雨绵绵的城市，但对于 AST 来说，这最好不过了，因为在这样的环境下他就可以经常待在家中摆弄他的计算机了。

MINIX 是他 1987 年编制的，主要用于学生学习操作系统原理。到 1991 年时版本是 1.5。目前主要有两个版本在使用：1.5 版和 2.0 版。当时该操作系统在大学使用是免费的，但其它用途则不是。当然目

前 MINIX 系统已经是免费的，可以从许多 FTP 上下载。

对于 Linux 系统，他后来曾表示对其开发者 Linus 的称赞。但他认为 Linux 的发展很大原因是由于他为了保持 MINIX 的小型化，能让学生在一个学期内就能学完，因而没有接纳全世界许多人对 MINIX 的扩展要求。因此在这样的前提下激发了 Linus 编写 Linux 系统。当然 Linus 也正好抓住了这个好时机。

作为一个操作系统，MINIX 并不是优秀者，但它同时提供了用 C 语言和汇编语言编写的系统源代码。这是第一次使得有抱负的程序员或 hacker 能够阅读操作系统的源代码。在当时，这种源代码是软件商们一直小心守护着的秘密。

1.1.3 GNU 计划

GNU 计划和自由软件基金会 FSF(the Free Software Foundation)是由 Richard M. Stallman 于 1984 年一手创办的。旨在开发一个类似 UNIX 并且是自由软件的完整操作系统：GNU 系统（GNU 是"GNU's Not Unix"的递归缩写，它的发音为"guh-NEW"）。各种使用 Linux 作为核心的 GNU 操作系统正在被广泛的使用。虽然这些系统通常被称作"Linux"，但是 Stallman 认为，严格地说，它们应该被称为 GNU/Linux 系统。

到上世纪 90 年代初，GNU 项目已经开发出许多高质量的免费软件，其中包括有名的 emacs 编辑系统、bash shell 程序、gcc 系列编译程序、gdb 调试程序等等。这些软件为 Linux 操作系统的开发创造了一个合适的环境。这是 Linux 能够诞生的基础之一，以至于目前许多人都将 Linux 操作系统称为“GNU/Linux”操作系统。

1.1.4 POSIX 标准

POSIX (Portable Operating System Interface for Computing Systems) 是由 IEEE 和 ISO/IEC 开发的一簇标准。该标准是基于现有的 UNIX 实践和经验，描述了操作系统的调用服务接口。用于保证编制的应用程序可以在源代码一级上在多种操作系统上移植和运行。它是在 1980 年早期一个 UNIX 用户组 (usr/group) 的早期工作基础上取得的。该 UNIX 用户组原来试图将 AT&T 的 System V 操作系统和 Berkeley CSRG 的 BSD 操作系统的调用接口之间的区别重新调和集成。并于 1984 年定制出了 /usr/group 标准。

1985 年，IEEE 操作系统技术委员会标准小组委员会 (TCOS-SS) 开始在 ANSI 的支持下责成 IEEE 标准委员会制定有关程序源代码可移植性操作系统服务接口正式标准。到了 1986 年 4 月，IEEE 制定出了试用标准。第一个正式标准是在 1988 年 9 月份批准的 (IEEE 1003.1-1988)，也既以后经常提到的 POSIX.1 标准。

到 1989 年，POSIX 的工作被转移至 ISO/IEC 社团，并由 15 工作组继续将其制定成 ISO 标准。到 1990 年，POSIX.1 与已经通过的 C 语言标准联合，正式批准为 IEEE 1003.1-1990 (也是 ANSI 标准) 和 ISO/IEC 9945-1:1990 标准。

POSIX.1 仅规定了系统服务应用程序编程接口 (API)，仅概括了基本的系统服务标准。因此工作组期望对系统的其它功能也制定出标准。这样 IEEE POSIX 的工作就开始展开了。刚开始有十个批准的计划在进行，有近 300 多人参加每季度为期一周的会议。着手的工作有命令与工具标准 (POSIX.2)、测试方法标准 (POSIX.3)、实时 API (POSIX.4) 等。到了 1990 年上半年已经有 25 个计划在进行，并且有 16 个工作组参与了进来。与此同时，还有一些组织也在制定类似的标准，如 X/Open, AT&T, OSF 等。

在 90 年代初，POSIX 标准的制定正处在最后投票敲定的时候，那是 1991-1993 年间。此时正是 Linux 刚刚起步的时候，这个 UNIX 标准为 Linux 提供了极为重要的信息，使得 Linux 能够在标准的指导下进行开发，并能够与绝大多数 UNIX 操作系统兼容。在最初的 Linux 内核源代码中 (0.01 版、0.11 版) 就已经为 Linux 系统与 POSIX 标准的兼容做好了准备工作。在 Linux 0.01 版内核的 /include/unistd.h 文件中就已经定义了几个有关 POSIX 标准要求的符号常数，而且 Linus 在注释中已写道：“OK，这也许是个玩笑，但我正在着手研究它呢”。

1991 年 7 月 3 日在 comp.os.minix 上发布的 post 上就已经提到了正在搜集 POSIX 的资料。其中透露了他正在着手一个操作系统的开发，并且在开发之初已经想到要实现与 POSIX 相兼容的问题了。

1.1.5 Linux 操作系统的诞生

在 1981 年，IBM 公司推出了享誉全球的微型计算机 IBM PC。在 1981-1991 年间，MS-DOS 操作系统一直是微型计算机系统的主宰。此时计算机硬件价格虽然逐年下降，但软件价格仍然居高不下。当时 Apple 的 MACs 操作系统可以说是性能最好的，但是其天价使得没人能够轻易靠近。

当时的另一个计算机技术阵营就是 UNIX 世界。但是 UNIX 操作系统就不仅是价格昂贵的问题了。为了寻求高利润率，UNIX 经销商们把价格抬得极高，PC 小用户根本不能靠近它。曾经一度收到 Bell Labs 许可而能在大学中用于教学的 UNIX 源代码也一直被小心地守卫着不许公开。对于广大的 PC 用户，软件行业的大型供应商们始终没有给出有效的解决这个问题的手段。

正在此时，出现了 MINIX 操作系统，并且有一本描述其设计实现原理的书同时发行。由于 AST 的这本书写的非常详细，并且叙述得有条有理，于是几乎全世界的计算机爱好者都开始看这本书，以期能理解操作系统的工作原理。其中也包括 Linux 系统的创始者 Linus Benedict Torvalds。

当时(1991 年)，Linus Benedict Torvalds 是赫尔辛基大学计算机科学系的二年级学生，也是一个自学的计算机 hacker。这个 21 岁的芬兰年轻人喜欢鼓捣他的计算机，测试计算机的性能和限制。但当时他所缺乏的就是一个专业级的操作系统。

在同一年间，GNU 计划已经开发出了许多工具软件。其中最受期盼的 GNU C 编译器已经出现，但还没有开发出免费的 GNU 操作系统。即使是教学使用的 MINIX 操作系统也开始有了版权，需要购买才能得到源代码。虽然 GNU 的操作系统 HURD 一直在开发之中，但在当时看来不能在几年内完成。

为了更好地学习计算机知识（或许也只是为了兴趣☺），Linus 使用圣诞节的压岁钱和贷款购买了一台 386 兼容电脑，并从美国邮购了一套 MINIX 系统软件。就在等待 MINIX 软件期间，Linus 认真学习了有关 Intel 80386 的硬件知识。为了能通过 Modem 拨号连接到学校的主机上，他使用汇编语言并利用 80386 CPU 的多任务特性编制出一个终端仿真程序。此后为了将自己一台老式电脑上的软件复制到新电脑上，他还为软盘驱动器、键盘等硬件设备编制出相应的驱动程序。

通过编程实践，并在学习过程中认识到 MINIX 系统的诸多限制（MINIX 虽然很好，但只是一个用于教学目的简单操作系统，而不是一个强有力的实用操作系统），而且通过上述实践 Linus 已经有了一些类似于操作系统硬件设备驱动程序的代码，于是他开始有了编制一个新操作系统的想法。此时 GNU 计划已经开发出许多工具软件，其中最受期盼的 GNU C 编译器已经出现。虽然 GNU 的免费操作系统 HURD 正在开发中。但 Linus 已经等不急了。

从 1991 年 4 月份起，他通过修改终端仿真程序和硬件驱动程序，开始编制起自己的操作系统来。刚开始，他的目的很简单，只是为了学习 Intel 386 体系结构保护模式运行方式下的编程技术。但后来 Linux 的发展却完全改变了初衷。根据 Linus 在 comp.os.minix 新闻组上发布的消息，我们可以知道他逐步从学习 MINIX 系统阶段发展到开发自己的 Linux 系统的过程。

Linus 第 1 次向 comp.os.minix 投递消息是在 1991 年 3 月 29 日。所发帖子的题目是“gcc on minix-386 doesn't optimize”，是有关 gcc 编译器在 MINIX-386 系统上运行优化的问题（MINIX-386 是一个由 Bruce Evans 改进的利用 Intel 386 特性的 32 位 MINIX 系统）。由此可知，Linus 在 1991 年初期就已经开始深入研究了 MINIX 系统，并在这段时间有了改进 MINIX 操作系统的思想。在进一步学习 MINIX 系统之后，这个想法逐步演变成想重新设计一个基于 Intel 80386 体系结构的新操作系统的构思。

他在回答有人提出 MINIX 上的一个问题时，所说的第一句话就是“阅读源代码”（“RTFSC (Read the F**ing Source Code :-)”）。他认为答案就在源程序中。这也说明了对于学习系统软件来说，我们不光需要懂得系统的工作基本原理，还需要结合实际系统，学习实际系统的实现方法。因为理论毕竟是理论，其中省略了许多枝节，而这些枝节问题虽然没有太多的理论含量，但却是一个系统必要的组成部分，就象麻雀身上的一根羽毛。

从 1991 年 4 月份开始, Linus 几乎花费了全部时间研究 MINIX-386 系统(Hacking the kernel), 并且尝试着移植 GNU 的软件到该系统上(GNU gcc、bash、gdb 等)。并于 4 月 13 日在 comp.os.minix 上发布说自己已经成功地将 bash 移植到了 MINIX 上, 而且已经爱不释手、不能离开这个 shell 软件了。

第一个与 Linux 有关的消息是在 1991 年 7 月 3 日在 comp.os.minix 上发布的(当然, 那时还不存在 Linux 这个名称, 当时 Linus 脑子里想的名称可能是 FREAX ☺, FREAX 的英文含义是怪诞的、怪物、异想天开等)。其中透露了他正在进行 Linux 系统的开发, 并且已经想到要实现与 POSIX 兼容的问题了。

在 Linus 另一个发布的消息中(1991 年 8 月 25 日 comp.os.minix), 他向所有 MINIX 用户询问 “What would you like to see in minix?” (“你最想在 MINIX 系统中见到什么?”), 在该消息中他首次透露出正在开发一个(免费的)386(486)操作系统, 并且说只是兴趣而已, 代码不会很大, 也不会象 GNU 的那样专业。希望大家反馈一些对于 MINIX 系统中喜欢哪些特色不喜欢什么等信息, 并且说明由于实际和其它一些原因, 新开发的系统刚开始与 MINIX 很象(并且使用了 MINIX 的文件系统)。并且已经成功地将 bash(1.08 版)和 gcc(1.40 版)移植到了新系统上, 而且在过几个月就可以实用了。

最后, Linus 申明他开发的操作系统没有使用一行 MINIX 的源代码; 而且由于使用了 386 的任务切换特性, 所以该操作系统不好移植(没有可移植性), 并且只能使用 AT 硬盘。对于 Linux 的移植性问题, Linus 当时并没有考虑。但是目前 Linux 几乎可以运行在任何一种硬件体系结构上。

到了 1991 年的 10 月 5 日, Linus 在 comp.os.minix 新闻组上发布消息, 正式向外宣布 Linux 内核系统的诞生(Free minix-like kernel sources for 386-AT)。这段消息可以称为 Linux 的诞生宣言, 并且一直广为流传。因此 10 月 5 日对 Linux 社区来说是一个特殊的日子, 许多后来 Linux 的新版本发布时都选择了这个日子。所以 RedHat 公司选择这个日子发布它的新系统也不是偶然的。

1.1.6 Linux 操作系统版本的变迁

Linux 操作系统从诞生到 1.0 版正式出现, 共发布了表 1-1 中所示的一些主要版本。

表 1-1 内核的主要版本

版本号	发布日期	说明
0.00	(1991.2-4)	两个进程, 分别在屏幕上显示'AAA'和'BBB'。
0.01	(1991.8)	第一个正式向外公布的 Linux 内核版本。多线程文件系统、分段和分页内存管理。
0.02	(1991.10.5)	该版本以及 0.03 版是内部版本, 目前已经无法找到特点同上。
0.10	(1991.10)	由 Ted Ts'o 发布的 Linux 内核版本。增加了内存分配库函数。
0.11	(1991.12.8)	基本可以正常运行的内核版本。至此硬盘和软驱驱动。
0.12	(1992.1.15)	主要增加了数学协处理器的软件模拟程序, 增加了作业控制、虚拟控制台、文件符号链接和虚拟内存对换功能。
0.95(0.13)	(1992.3.8)	加入虚拟文件系统支持, 增加了登录功能。改善了软盘驱动程序和文件系统的性能。改变了硬盘编号方式。支持 CDROM。
0.96	(1992.5.12)	开始加入网络支持。改善了串行驱动、高速缓冲、内存管理的性能, 支持动态链接库, 并能运行 X-Windows 程序。
0.97	(1992.8.1)	增加了对新的 SCSI 驱动程序的支持。
0.98	(1992.9.29)	改善了对 TCP/IP (0.8.1) 网络的支持, 纠正了 extfs 的错误。
0.99	(1992.12.13)	重新设计进程对内存的使用分配, 每个进程有 4G 线性空间。
1.0	(1994.3.14)	第一个正式版。

将 Linux 系统 0.13 版内核直接改称 0.95 版, Linus 的意思是让大家不要觉得离 1.0 版还很遥远。同时, 从 0.95 版开始, 对内核的许多改进之处(补丁程序的提供)均以其他人为主了, 而 Linus 的主要任务开始变成对内核的维护和决定是否采用某个补丁程序。到现在为止, 最新的内核版本是 2003 年 12 月 18 日公布的 2.6.2 版。其中包括大约 15000 个文件, 而且使用 gz 压缩后源代码软件包也有 40MB 左右! 到现在为止, 最新版见表 1-2 所示。

表 1-2 新内核源代码字节数

内核版本号	发布日期	源代码大小(经 gz 压缩后)
2.4.22	2004.2.4	35MB
2.6.5	2004.4.4	41MB

1.1.7 Linux 名称的由来

Linux 操作系统刚开始时并没有被称作 Linux, Linus 给他的操作系统取名为 FREAX, 其英文含义是怪诞的、怪物、异想天开等意思。在他将新的操作系统上载到 ftp.funet.fi 服务器上时, 管理员 Ari Lemke 很不喜欢这个名称。他认为既然是 Linus 的操作系统就取其谐音 Linux 作为该操作系统的目录吧, 于是 Linux 这个名称就开始流传下来。

在 Linus 的自传《Just for Fun》一书中, Linus 解释说¹:

“坦白地说, 我从来没有想到过要用 Linux 这个名称发布这个操作系统, 因为这个名字有些太自负了。而我为最终发布版准备的是什么呢? Freax。实际上, 内核代码中某些早期的 Makefile - 用于描述如何编译源代码的文件 - 文件中就已经包含有“Freax”这个名字了, 大约存在了半年左右。但其实这也没什么关系, 在当时还不需要一个名字, 因为我还没有向任何人发布过内核代码。”

“而 Ari Lemke, 他坚持要用自己的方式将内核代码放到 ftp 站点上, 并且非常不喜欢 Freax 这个名字。他坚持要用现在这个名字(Linux), 我承认当时我并没有跟他多争论。但这都是他取的名字。所以我可以光明正大地说我并不自负, 或者部分坦白地说我并没有本位主义思想。但我想好吧, 这也是个好名字, 而且以后为这事我总能说服别人, 就象我现在做的这样。”

1.1.8 早期 Linux 系统开发的主要贡献者

从 Linux 早期源代码中可以看出, Linux 系统的早期主要开发人员除了 Linus 本人以外, 最著名的人员之一就是 Theodore Ts'o (Ted Ts'o)。他于 1990 年毕业于 MIT 计算机专业。在大学时代他就积极参加学校中举办的各种学生活动。他喜欢烹饪、骑自行车, 当然还有就是 Hacking on Linux。后来他开始喜欢起业余无线电报运动。目前他在 IBM 工作从事系统编程及其它重要事务。他还是国际网络设计、操作、销售和研究者开放团体 IETF 成员。

Linux 在世界范围内的流行也有他很大的功劳。早在 Linux 操作系统刚问世时, 他就怀着极大的热情为 linux 的发展提供了 Maillist, 几乎是在 Linux 刚开始发布时起, 他就一直在为 Linux 做出贡献。他也是最早向 Linux 内核添加程序的人(Linux 内核 0.10 版中的虚拟盘驱动程序 ramdisk.c 和内核内存分配程序 kmalloc.c)。直到目前为止他仍然从事着与 Linux 有关的工作。在北美洲地区他最早设立了 Linux 的 ftp 站点 (tsx-11.mit.edu), 而且该站点至今仍然为广大 Linux 用户提供服务。他对 Linux 作出的最大贡献之一是提出并实现了 ext2 文件系统。该文件系统现已成为 Linux 世界中事实上的文件系统标准。最近他又推出了 ext3 文件系统。该系统大大提高了文件系统的稳定性和访问效率。作为对他的推崇, 第 97 期(2002 年 5 月)的 Linux Journal 期刊将他作为封面人物, 并对他进行了采访。目前, 他为 IBM Linux

¹ Linus Torvalds 《Just for fun》第 84-88 页。

技术中心工作，并从事着有关 Linux 标准库 LSB(Linux Standard Base)等方面的工作。

Linux 社区中另一位著名人物是 Alan Cox。他原工作于英国威尔士斯旺西大学(Swansea University College)。刚开始他特别喜欢玩电脑游戏，尤其是 MUD (Multi-User Dungeon or Dimension，多用户网络游戏)。在 90 年代早期 games.mud 新闻组的 posts 中你可以找到他发表的大量帖子。他甚至为此还写了一篇 MUD 的发展史(rec.games.mud 新闻组，1992 年 3 月 9 日，A history of MUD)。

由于 MUD 游戏与网络密切相关，慢慢地他开始对计算机网络着迷起来。为了玩游戏并提高电脑运行游戏的速度以及网络传输速度，他需要选择一个最为满意的操作平台。于是他开始接触各种类型的操作系统。由于没钱，即使是 MINIX 系统他也买不起。当 Linux 0.11 和 386BSD 发布时，他考虑良久总算购置了一台 386SX 电脑。由于 386BSD 需要数学协处理器支持，而采用 Intel 386SX CPU 的电脑是不带数学协处理器的，所以他安装了 Linux 系统。于是他开始学习带有免费源代码的 Linux，并开始对 Linux 系统产生了兴趣，尤其是有关网络方面的实现。在关于 Linux 单用户运行模式问题的讨论中，他甚至赞叹 Linux 实现得巧妙(beautifully)。

Linux 0.95 版发布之后，他开始为 Linux 系统编写补丁程序（修改程序）（记得他最早的两个补丁程序，都没有被 Linus 采纳），并成为 Linux 系统上 TCP/IP 网络代码的最早使用人之一。后来他逐渐加入了 Linux 的开发队伍，并成为维护 Linux 内核源代码的主要负责人之一，也可以说成为 Linux 社团中继 Linus 之后最为重要的人物。以后 Microsoft 公司曾经邀请他加盟，但他却干脆地拒绝了。从 2001 年开始，他负责维护 Linux 内核 2.4.x 的代码。而 Linus 主要负责开发最新开发版内核的研制(奇数版，比如 2.5.x 版)。

《内核黑客手册》(The Linux Kernel Hackers' Guide)一书的作者 Michael K. Johnson 也是最早接触 Linux 操作系统的人之一(从 0.97 版)。他还是著名 Linux 文档计划 (Linux Document Project - LDP) 的发起者之一。曾经在 Linux Journal 工作，现在 RedHat 公司工作。

Linux 系统并不是仅有这些中坚力量就能发展成今天这个样子的，还有许多计算机高手对 Linux 做出了极大的贡献，这里就不一一列举了。主要贡献者的具体名单可参见 Linux 内核中的 CREDITS 文件，其中以字母顺序列出了对 Linux 做出较大贡献的近 400 人的名单列表，包括他们的 email 地址和通信地址、主页以及主要贡献事迹等信息。

通过上述说明，我们可以对上述 Linux 的五大支柱归纳如下：

UNIX 操作系统 -- UNIX 于 1969 年诞生在 Bell 实验室。Linux 就是 UNIX 的一种克隆系统。UNIX 的重要性就不用多说了。

MINIX 操作系统 -- MINIX 操作系统也是 UNIX 的一种克隆系统，它于 1987 年由著名计算机教授 Andrew S. Tanenbaum 开发完成。由于 MINIX 系统的出现并且提供源代码(只能免费用于大学内)在全世界的大学中刮起了学习 UNIX 系统旋风。Linux 刚开始就是参照 MINIX 系统于 1991 年才开始开发。

GNU 计划-- 开发 Linux 操作系统，以及 Linux 上所用大多数软件基本上都出自 GNU 计划。Linux 只是操作系统的一个内核，没有 GNU 软件环境(比如说 bash shell)，则 Linux 将寸步难行。

POSIX 标准 -- 该标准在推动 Linux 操作系统以后朝着正规路上发展起着重要的作用。是 Linux 前进的灯塔。

INTERNET -- 如果没有 Internet 网，没有遍布全世界的无数计算机黑客的无私奉献，那么 Linux 最多只能发展到 0.13(0.95)版的水平。

1.2 内容综述

本文将主要对 Linux 的早期内核 0.11 版进行详细描述和注释。Linux-0.11 版本是在 1991 年 12 月 8

日发布的。在发布时包括以下文件：

bootimage.Z	- 具有美国键盘代码的压缩启动映像文件；
rootimage.Z	- 以 1200kB 压缩的根文件系统映像文件；
linux-0.11.tar.Z	- 内核源代码文件。大小为 94KB，展开后也仅有 325KB；
as86.tar.Z	- Bruce Evans' 二进制执行文件。是 16 位的汇编程序和装入程序；
INSTALL-0.11	- 更新过的安装信息文件。

目前除了原来的 `rootimage.Z` 文件，其它四个文件均能找到。不过作者已经利用 Internet 上的资源为 Linux 0.11 重新制作出了一个完全可以使用的 `rootimage-0.11` 根文件系统。并重新为其编译出能在 0.11 环境下使用的 gcc 1.40 编译器，配置出可用的实验开发环境。目前，这些文件均可以从 oldlinux.org 网站上下载。

本文主要详细分析 linux-0.11 内核中的所有源代码程序，对每个源程序文件都进行了详细注释，包括对 Makefile 文件的注释。分析过程主要是按照计算机启动过程进行的。因此分析的连贯性到初始化结束内核开始调用 shell 程序为止。其余的各个程序均针对其自身进行分析，没有连贯性，因此可以根据自己的需要进行阅读。但在分析时还是提供了一些应用实例。










所有的程序在分析过程中如果遇到作者认为是较难理解的语句时，将给出相关知识的详细介绍。比如，在阅读代码时一次遇到 C 语言内嵌汇编码时，将对 GNU C 语言的内嵌汇编语言进行较为详细的介绍；在遇到对中断控制器进行输入/输出操作时，将对 Intel 中断控制器（8259A）芯片给出详细的说明，并列出使用的命令和方法。这样做有助于加深对代码的理解，又能更好的了解所用硬件的使用方法，作者认为这种解读方法要比单独列出一章内容来总体介绍硬件或其它知识要效率高得多。

拿 Linux 0.11 版内核来“开刀”是为了提高我们认识 Linux 运行机理的效率。Linux-0.11 版整个内核源代码只有 325K 字节左右，其中包括的内容基本上都是 Linux 的精髓。而目前最新的 2.5.XX 版内核非常大，将近有 188 兆字节，即使你花一生的经历来阅读也未必能全部都看完。也许你要问“既然要从简入手，为什么不分析更小的 Linux 0.01 版内核源代码呢？它只有 240K 字节左右”主要原因是因为 0.01 版的内核代码有太多的不足之处，甚至还没有包括对软盘的驱动程序，也没有很好地涉及数学协处理器的使用以及对登陆程序的说明。并且其引导启动程序的结构也与目前的版本不太一样，而 0.11 版的引导启动程序结构则与现在的基本上是一样的。另外一个原因是可以找到 0.11 版早期的已经编译制作好的内核映像文件(`bootimage`)，可以用来进行引导演示。如果再配上简单的根文件系统映像文件(`rootimage`)，那么它就可以进行正常的运行了。

拿 Linux 0.11 版进行学习也有不足之处。比如该内核版本中尚不包括有关专门的进程等待队列、TCP/IP 网络等方面的一些当前非常重要的代码，对内存的分配和使用与现今的内核也有所区别。但好在 Linux 中的网络代码基本上是自成一体的，与内核机制关系不是非常大，因此可以在了解了 Linux 工作的基本原理之后再去分析这些代码。

本文对 Linux 内核中所有的代码都进行了说明。为了保持结构的完整性，对代码的说明是以内核中源代码的组成结构来进行的，基本上是以每个源代码中的目录为一章内容进行介绍。介绍的源程序文件的次序可参见前面的文件列表索引。整个 Linux 内核源代码的目录结构如下列表 1.1 所示。所有目录结构均是以 linux 为当前目录。

列表 1-1 Linux/目录

	名称	大小	最后修改日期(GMT)	说明
	boot/		1991-12-05 22:48:49	
	fs/		1991-12-08 14:08:27	
	include/		1991-09-22 19:58:04	
	init/		1991-12-05 19:59:04	
	kernel/		1991-12-08 14:08:00	
	lib/		1991-12-08 14:10:00	
	mm/		1991-12-08 14:08:21	
	tools/		1991-12-04 13:11:56	
	Makefile	2887 bytes	1991-12-06 03:12:46	

本书内容可以分为三个部分。第 1 章至第 4 章是描述内核引导启动和 32 位运行方式的准备阶段，作为学习内核的初学者应该全部进行阅读。第二部分从第 5 章到第 10 章是内核代码的主要部分。其中第 5 章内容可以作为阅读本部分后续章节的索引来进行。第 11 章到第 13 章是第三部分内容，可以作为阅读第二部分代码的参考信息。

第 2 章概要地描述了 Linux 操作系统的体系结构、内核源代码文件放置的组织结构以及每个文件大致功能。还介绍了 Linux 对物理内存的使用分配方式、内核的几种堆栈及其使用方式和虚拟线性地址的使用分配。最后开始注释内核程序包中 Linux/目录下的所看到的第一个文件，也即内核代码的总体 Makefile 文件的内容。该文件是所有内核源程序的编译管理配置文件，供编译管理工具软件 make 使用。

第 3 章将详细注释 boot/目录下的三个汇编程序，其中包括磁盘引导程序 bootsect.s、获取 BIOS 中参数的 setup.s 汇编程序和 32 位运行启动代码程序 head.s。这三个汇编程序完成了把内核从块设备上引导加载到内存的工作，并对系统配置参数进行探测，完成了进入 32 位保护模式运行之前的所有工作。为内核系统执行进一步的初始化工作做好了准备。

第 4 章主要介绍 init/目录中内核系统的初始化程序 main.c。它是内核完成所有初始化工作并进入正常运行的关键地方。在完成了系统所有的初始化工作后，创建了用于 shell 的进程。在介绍该程序时将需要查看其所调用的其它程序，因此对后续章节的阅读可以按照这里调用的顺序进行。由于内存管理程序的函数在内核中被广泛使用，因此该章内容应该最先选读。当你能真正看懂直到 main.c 程序为止的所有程序时，你应该已经对 Linux 内核有了一定的了解，可以说已经有一半入门了☺，但你还需要对文件系统、系统调用、各种驱动程序等进行更深一步的阅读。

第 5 章主要介绍 kernel/目录中的所有程序。其中最重要的部分是进程调度函数 schedule()、sleep_on() 函数和有关系统调用的程序。此时你应该已经对其中的一些重要程序有所了解。

第 6 章对 kernel/dev_blk/目录中的块设备程序进行了注释说明。该章主要含有硬盘、软盘等块设备的驱动程序，主要用来与文件系统和高速缓冲区打交道，含有较多与硬件相关的内容。因此，在阅读这章内容时需参考一些硬件资料。最好能首先浏览一下文件系统的章节。

第 7 章对 kernel/dev_chr/目录中的字符设备驱动程序进行注释说明。这一章中主要涉及串行线路驱动程序、键盘驱动程序和显示器驱动程序。这些驱动程序构成了 0.11 内核支持的串行终端和控制台终端设备。因此本章也含有较多与硬件有关的内容。在阅读时需要参考一下相关硬件的书籍。

第 8 章介绍 kernel/math/目录中的数学协处理器的仿真程序。由于本书所注释的内核版本，还没有真正开始支持协处理器，因此本章的内容较少，也比较简单。只需有一般性的了解即可。

第 9 章介绍内核源代码 fs/目录中的文件系统程序，在看这章内容时建议你能够暂停一下而去阅读 Andrew S. Tanenbaum 的《操作系统设计与实现》一书中有关 MINIX 文件系统的章节，因为最初的 Linux 系统是只支持 MINIX 一种文件系统，Linux 0.11 版也不例外。

第 10 章解说 `mm/` 目录中的内存管理程序。要透彻地理解这方面的内容，需要对 Intel 80X86 微处理器的保护模式运行方式有足够的理解，因此本章在适当的地方包含有较为完整的有关 80X86 保护模式运行方式的说明，这些知识基本上都可以参考 Intel 80386 程序员编程手册(Intel 80386 Programmer's Reference Manual)。但在此章中，以源代码中的运用实例为对象进行解说，应该可以更好地理解它的工作原理。

现有的 Linux 内核分析书籍都缺乏对内核头文件的描述，因此对于一个初学者来讲，在阅读内核程序时会碰到许多障碍。本书的第 11 章对 `include/` 目录中的所有头文件进行了详细说明，基本上对每一个定义、每一个常量或数据结构都进行了详细注释。为了便于在阅读时参考查阅，本书在附录中还对一些经常要用到的重要的数据结构和变量进行了归纳注释，但这些内容实际上都能在这一章中找到。虽然该章内容主要是为阅读其它章节中的程序作参考使用的，但是若想彻底理解内核的运行机制，仍然需要了解这些头文件中的许多细节。

第 12 章介绍了 Linux 0.11 版内核源代码 `lib/` 目录中的所有文件。这些库函数文件主要向编译系统等系统程序提供了接口函数，对以后理解系统软件会有较大的帮助。由于这个版本较低，所以这里的内容并不是很多，可以很快地看完。这也是我们为什么选择 0.11 版的原因之一。

第 13 章介绍 `tools/` 目录下的 `build.c` 程序。这个程序并不会包括在编译生成的内核映像(image)文件中，它仅用于将内核中的磁盘引导程序块与其它主要内核模块连接成一个完整的内核映像(kernel image)文件。

第 14 章介绍了学习内核源代码时的实验环境以及实验方法。主要介绍了在 Bochs 仿真系统下使用和编译 Linux 内核的方法以及磁盘镜象文件的制作方法。还说明了如何修改 Linux 0.11 源代码的语法使其能在 RedHat 9 系统下顺利编译出正确的内核来。

最后是附录和索引。附录中给出了 Linux 内核中的一些常数定义和基本数据结构定义，以及保护模式运行机制的简明描述。

为了便于查阅，在本书的附录中还单独列出了内核中要用到的有关 PC 机硬件方面的信息。在参考文献中，我们仅给出了在阅读源代码时可以参考的书籍、文章等信息，并没有包罗万象地给出一大堆的繁杂凌乱的文献列表。比如在引用 Linux 文档项目 LDP (Linux Document Project) 中的文件时，我们会明确地列出具体的需要参考哪一篇 HOWTO 文章，而并不是仅仅给出 LDP 的网站地址了事。

Linus 在最初开发 Linux 操作系统内核时，主要参考了 3 本书。一本是 M. J. Bach 著的《UNIX 操作系统设计》，该书描述了 UNIX System V 内核的工作原理和数据结构。Linus 使用了该书中很多函数的算法，Linux 内核源代码中很多重要函数的名称都取自该书。因此，在阅读本书时，这是一本必不可少的内核工作原理方面的参考书籍。另一本是 John H. Crawford 等编著的《Programming the 80386》，是讲解 80x86 下保护模式编程方法的好书。还有一本就是 Andrew S. Tanenbaum 著的《MINIX 操作系统设计与实现》一书第 1 版。Linus 主要使用了该书中描述的 MINIX 文件系统 1.0 版，而且在早期的 Linux 内核中也仅支持该文件系统，所以在阅读本书有关文件系统一章内容时，文件系统的工作原理方面的知识完全可以从 Tanenbaum 的书中获得。

在对每个程序进行解说时，我们首先简单说明程序的主要用途和目的、输入输出参数以及与其它程序的关系，然后列出程序的完整代码并在其中对代码进行详细注释，注释时对原程序代码或文字不作任何方面的改动或删除，因为 C 语言是一种英语类语言，程序中原有的少量英文注释对常数符号、变量名等也提供了不少有用的信息。在代码之后是对程序更为深入的解剖，并对代码中出现的一些语言或硬件方面的相关知识进行说明。如果在看完这些信息后回头再浏览一遍程序，你会有更深一层的体会。

对于阅读本书所需要的一些基本概念知识的介绍都散布在各个章节相应的地方，这样做主要是为了能够方便的找到，而且在结合源代码阅读时，对一些基本概念能有更深的理解。

最后要说明的是当你已经完全理解了本文所解说的一切时，并不代表你已经成为一个 Linux 行家了，

你只是刚刚踏上 Linux 的征途，具有了一定的成为一个 Linux GURU 的初步知识。这时你应该去阅读更多的源代码，最好是循序渐进地从 1.0 版本开始直到最新的正在开发中的奇数编号的版本。在撰写这本书时最新的 Linux 内核是 2.5.44 版。当你能快速理解这些开发中的最新版本甚至能提出自己的建议和补丁（patch）程序时，我也甘拜下风了☺。

1.3 本章小结

首先阐述了 Linux 诞生和发展不可缺少的五个支柱：UNIX 最初的开放源代码版本为 Linux 提供了实现的基本原理和算法、Richard Stallman 的 GNU 计划为 Linux 系统提供了丰富且免费的各种实用工具、POSIX 标准的出现为 Linux 提供了实现与标准兼容系统的参考指南、A.S.T 的 MINIX 操作系统为 Linux 的诞生起到了不可忽缺的参考、Internet 是 Linux 成长和壮大的必要环境。最后本章概述了书中的基本内容。