TURING 图灵程序设计丛书 Linux/Unix系列

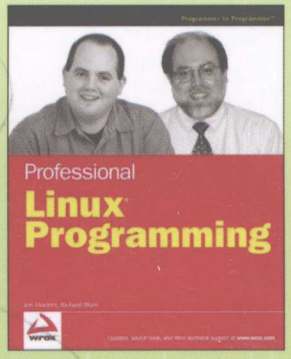
wrox

Professional Linux Programming

Linux 高级程序设计

[英] Jon Masters
[美] Richard Blum 著
陈健 等 译

- 畅销书《Linux 程序设计（第3版）》后使你更上一层楼的经典著作
- 著名开源技术社区LUPA强烈推荐
- 全面阐述现代Linux程序设计的技术和方法



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Linux/Unix系列

Professional Linux Programming

Linux 高级程序设计

[英] Jon Masters
[美] Richard Blum 著
陈健 等 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

Linux 高级程序设计 /（英）美斯特（Masters, J.），（美）布卢（Blum,R.）著；陈健等译．—北京：人民邮电出版社，2008.7
（图灵程序设计丛书）
书名原文：Professional Linux Programming
ISBN 978-7-115-17910-4

Ⅰ．L… Ⅱ．①美…②布…③陈… Ⅲ．Linux 操作系统—程序设计 Ⅳ．TP316.89

中国版本图书馆 CIP 数据核字（2008）第 046781 号

内 容 提 要

本书是 Linux 程序设计领域的一部力作，讲解了大量程序员需要掌握的关键知识点，包括 Linux 开发中的基本工具、Linux 系统编程、Linux 桌面开发以及 Linux 与 Web 开发。书中包括大量有益的经验之谈和富于启发的示例。

本书主要针对已有一定 Linux 开发经验或者从其他平台转到 Linux 平台的专业程序员，同样也适合想更多了解系统以解决实际问题的 Linux 使用者。

图灵程序设计丛书

Linux 高级程序设计

◆ 著　［英］Jon Masters　［美］Richard Blum
译　陈　健　等
责任编辑　杨福川

◆ 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　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷
新华书店总店北京发行所经销

◆ 开本：800×1000　1/16
印张：25.25
字数：660 千字　2008 年 7 月第 1 版
印数：1－4 000 册　2008 年 7 月河北第 1 次印刷
著作权合同登记号　图字：01-2007-0946 号

ISBN 978-7-115-17910-4/TP

定价：59.00 元

读者服务热线：(010)88593802　印装质量热线：(010)67129223
反盗版热线：(010)67171154

版 权 声 明

译 者 序

与市面上其他的Linux编程类图书不同，本书是一本专门介绍Linux最新编程技术的图书，它没有像其他图书那样介绍很多Linux和其他类UNIX系统共有的内容，而是专门为Linux程序员所撰写的。

本书涉及面非常广，从基本工具和技术的使用到LAMP技术的介绍，从系统底层内核的剖析到网络、数据库编程，从GNOME桌面环境到图形、音频和游戏编程，可以说涵盖了现代Linux编程的各个方面。作者使用简炼的笔法勾勒出了Linux编程的全景。我在翻译本书时深深为作者在Linux方面的广博学识和深刻见解所折服。

由于本书涉及面非常广泛而篇幅又不长，所以作者采用的是技术概述、要点介绍、示例阐述的方式来组织本书的内容，对每一方面的内容的最新发展动态和技术要点都进行了介绍，并给出了进一步学习和研究所需要参考的内容的链接，但并不进行事无巨细的阐述，要求读者在阅读本书时在感兴趣的内容上参考大量其他的资料以进行进一步的学习。对如今这个信息极为丰富的时代来说，读者缺乏的并不是技术资料和文档，而是如何选择这些资料和更有效地进行学习以跟上Linux编程的最新技术进展，相信本书会为读者打开通向现代Linux编程的大门。

在翻译过程中我对原书中的一些明显错误进行了更正，为了帮助读者理解，还对一些名词、概念进行了译注，但因为本书涉及面非常广，有些领域也并非我所擅长，所以译文中的错误在所难免，真诚地希望读者能提出宝贵意见，以便在本书重印时进行修订。

最后，感谢浙江Linux专业委员会副主任兼著名开源技术社区LUPA（www.lupaworld.com）的负责人邵炜先生的强烈推荐。同时，还要感谢人民邮电出版社的编辑，正是因为他们始终如一的鼓励和支持，本书的翻译工作才得以顺利地完成。

陈 健

2007年冬于南京大学

前　言

Linux近几年来有了很大的发展，已从一个不起眼的小玩意发展到在越来越多的《财富》500强公司中发挥巨大作用。从人们使用的手机到最大型的超级计算机集群，几乎都在使用Linux内核和为Linux编译的软件。但究竟什么是Linux？是什么使得它和目前市场上其他的类UNIX操作系统区别开来的呢？最重要的是，如何才能在我们的软件项目中充分利用Linux的强大功能和广泛使用的自由/开放源码软件（Free，Libre，and Open Source Software，简称FLOSS[①]）带来的变革呢？

本书的目的就是为了讨论这些问题以及其他问题。写作本书的目的源自于读者的这样一种需求，即究竟是什么使得Linux如此独一无二，但本书并不是一本适合Linux初学者的指南，因为这样的书早已在市场上存在了。这些年来，作为一位专业的Linux程序员，我们发现一起工作的很多技术精湛的软件工程师都缺乏或没有Linux编程方面的经验。其中一些工程师一直在寻找与本书类似的图书，但最后总是失望而归。为了让读者不再遭受这样的挫折，本书将帮助读者理解Linux社区的强大意义、已确立的软件开发模型和Linux世界中处理事务的方式。

有许多图书声称是专为Linux编程而写的，其中有许多书确实非常出色，但它们往往过于集中地介绍Linux简单继承自其前辈的内容。在本书中你不会发现这些内容，本书不是一本只介绍Linux和其他老版本UNIX系统共有内容的图书，而是一本介绍现代Linux操作系统的图书。本书不仅仅是另外一本UNIX编程类图书，它试图解释为什么Linux这么成功，并向读者展示在这个主题上被其他图书一笔带过或完全忽略的系统中的某些部分。

在本书中，你将学习到是什么推动了Linux的开发过程。你将了解各种各样常被Linux开发人员使用的工具——编译器、调试器和软件配置管理工具，以及这些工具是如何用来构建应用软件、工具甚至Linux内核自身的。你将学习到Linux系统中使其与其他类UNIX系统真正区分开来的特有组件，你还将深入研究Linux系统的内部工作机理，以便更好地理解作为新一代Linux开发人员你所需要扮演的角色。

你将学习一些新颖的开发方法，包括虚拟化技术的使用和交叉编译的使用（一种为不同的兼容平台编译软件的手段）。你还将学习对于一个没有国界的社区来说软件国际化的重要性——Linux是真正国际性的，它的用户也是如此。最后，你将通过为热门的LAMP（Linux、Apache、MySQL、Perl/Python）组合编写软件来学习Linux在现代因特网上的广泛用途。Linux所包含的内容远不只是Linux内核，作为一位Linux开发人员，意识到这一点是非常重要的。

最重要的是，本书将为未来进一步学习打下基础。通过对推动Linux开发的关键主题的深刻讨论，我们将为你打开通向自由/开放源码软件项目世界的大门。在阅读本书之后，你将能更好地明白你究竟

① 2000年，Rishab Ghosh在荷兰创造了FLOSS这个词，libre是法语中的“自由”一词。——译者注

需要了解什么，你并不会在本书中找到所有的答案，但你将具备自己发现这些答案的能力。不论你是使用Linux编写自由软件还是参与一个大型商业软件项目，你都将在阅读本书中有所收获。

读者对象

本书为两类不同的读者服务。首先，本书面向的是准备转向Linux开发平台的程序员，这类读者已经熟悉C编程语言，并理解了编译器、链接器和调试器等基本概念。他们有可能已看过这方面的介绍性图书，例如，Wrox的*Beginning Linux Programming*（Wiley 2004）[①]，但却缺乏实践经验。

对于那些从事专业Linux软件开发的新手而言，本书的内容安排非常有利于学习。你可以按顺序逐章阅读全书，也可以有选择地跨过与内核相关的章节（第7～9章），而把学习重点放在在每天的项目中都会用到的与更高层次应用程序和工具相关的章节。你还将在本书中找到工具链（Toolchain）、可移植性和有特定用途的SCM（Software Configuration Management，软件配置管理）的背景知识。

对于那些已在他们的日常生活中使用Linux并且想要深入了解一个典型的Linux系统的内部工作原理，而又不需要开发软件的Linux爱好者、管理人员和其他有关人员来说，本书也包含他们感兴趣的内容。现代Linux系统是如何进行硬件检测的？为什么Linux内核不提供设备驱动程序模块？Linux是如何支持国际化的？许多类似的问题都可以在本书中找到答案。

对于那些已经有Linux使用经验的读者来说，并不需要阅读本书的全部内容，但可能也会在每一章中发现一些新鲜和有趣的内容。我们通常会在脚注和注释中包括一些你可能在以前没有遇到过的示例和建议，其中包括从其他人的经验中获取的轶事和教训。你可能会选择投入更多的精力在本书后面讨论Linux内核、桌面和LAMP的章节中。

最后，不管你是一位对Linux或UNIX有基本了解同时又希望开扩视野的微软的Windows开发人员，还是一位从过去的岁月走过来的执着的UNIX程序员，希望了解是什么使得Linux如此成功，本书都会对你有所帮助。

主要内容

本书涵盖了各种各样用于Linux软件开发和软件本身的技术，包括现代UNIX、类UNIX和Linux系统的背景知识，从一个平台到另一个平台的软件可移植性以及有助于在现代的Linux软件发行中实现这一目标的工具。你将学习到如何通过网络接口、图形化用户环境、复杂的基于Web的现代LAMP组合来与Linux系统进行交互，甚至将学习到如何扩展Linux内核本身。在本书中你将学习到的是现代Linux的开发技术。

本书的内容反映了写作时技术的最新发展水平，但软件的版本却在不断变化。因此，本书讨论的大多数主题并不要求使用某个特定版本的工具、源代码或发行包。如果需要，我们会在书中指出，否则，你可以假设书中的示例可以运行在任一个你所使用的最新的Linux发行版本上。

组织结构

本书大致分为4个部分。在第一部分中，你将了解一些基本的工具和技术，目的是让你（作为一

① 中文版《Linux程序设计》（第3版）已由人民邮电出版社出版。——编者注

位专业的Linux程序员）的生活更加轻松。你将了解GNU工具链、软件可移植的重要性和软件国际化的需求，以及许多其他的主题，这些内容能帮你提高软件项目开发的效率。你可能想先阅读这些内容，并会经常查阅它。

本书的第二部分介绍了一个典型的Linux系统的底层部分，即传统的系统编程的主题，包括网络、数据库概念和Linux内核。可以通过阅读这些内容来更好地理解你所感兴趣的主题，但你并不需要学习这些章节中的所有内容，特别是本书中关于Linux内核部分的内容，因为本书并不是一本Linux内核编程类的图书，但通过对这些内容的学习确实会让你更想继续深入学习。

在本书的第三部分中，你将了解一些更高级的概念，如GNOME桌面环境及其数量庞大的软件库。你将学习自由桌面项目（Free Desktop Project），并有机会利用Gstreamer库的强大功能编写一个简单的CD播放器应用程序，该函数库被用于现代GNOME桌面多媒体应用程序的开发。你将会发现有多少代码可以通过软件复用来实现，并获得一些编写自己的GNOME软件的深刻体会。

本书的最后一章专门介绍LAMP。通过基于商品化的软件栈并使用Linux、Apache、MySQL和Perl/Python来编译，LAMP允许你只使用自由开放源码的软件就可以编写功能非常强大的Web应用程序。该章将介绍这些组件中的每一部分并提供一些使用示例。

排版约定

为了帮助您更好地理解本书的内容并清楚发生了什么，我们贯穿全书使用了一些排版约定。

像这样的文本框用于放置重要的、不应被忘记的信息，这些信息和上下文的内容直接相关。

与当前讨论内容有关的技巧、提示、诀窍会单独列出并以楷体显示。

文本的格式如下所示。

- 在介绍新的术语和重要的词汇时以**黑体字**显示它们。
- 以类似“Ctrl+A”的格式来显示组合键。
- 以类似`persistence.properties`的格式来显示文本中的文件名、URL和代码。
- 以两种不同的方式来显示代码：

```
在代码示例中，以灰色背景来高亮显示新的和重要的代码。
```

对当前上下文并不重要的或已经显示过的代码不会以灰色高亮显示。

源代码

当你阅读本书中的示例时，可能会选择手工敲入所有代码或使用随本书附带的源代码文件。本书中使用的所有源代码都可以通过网址http://www.wrox.com下载[①]。访问该网址后，只需简单地定位本书的书名（使用搜索栏或书名列表），然后点击图书详细页面中的Download Code（下载代码）链接就可以获得本书的所有源代码。

① 源代码也可以在图灵网站www.turing book.com本书网页免费注册下载。——编者注

下载了源代码之后，只需使用你所喜欢的压缩工具解压缩它即可。此外，你也可以访问Wrox的主代码下载页面http://www.wrox.com/dynamic/books/download.aspx来查看本书的代码和所有Wrox出版的其他图书的代码。

勘误

我们已尽最大努力来确保本书的文字或代码中没有错误。然而，没有人是完美的，错误在所难免。如果你发现了本书中的错误，如拼写错误或错误的代码段，我们将非常感谢您的反馈意见。通过向我们发送勘误表，你不仅节省了其他读者困惑的时间，同时也有助于我们提供更高质量的信息。

要找到本书的勘误页，请访问http://www.wrox.com并使用搜索栏或书名列表来定位本书。然后，在本书的详细页面中，点击Book Errata（图书勘误）链接。在勘误页中，你将会看到由Wrox编辑发表的关于本书的所有勘误。一个包括每本图书勘误表的完整图书列表也可以通过网址www.wrox.com/misc-pages/booklist.shtml访问到。

如果在本书的勘误页中没有发现你所找到的错误，你可以通过访问网址www.wrox.com/contact/techsupport.shtml并填写上面的表格来向我们发送你所找到的错误。我们将核查该信息，如果它是正确的，我们就会发送消息到本书的勘误页并在本书的后续版本中更正该问题。

p2p.wrox.com

如果你希望和本书的作者以及其他人进行讨论，请加入p2p.wrox.com上的P2P论坛。这个论坛是一个基于Web的系统，你可以在上面发表与Wrox图书以及相关技术有关的文章，并与其他读者和技术人员进行交流。这个论坛提供了订阅功能，当有新的文章发表时，论坛会将你所选择的感兴趣的主题通过E-mail发送给你。Wrox的作者、编辑、其他的行业专家和读者都加入了这些论坛。

你将会在网址http://p2p.wrox.com上发现许多不同的论坛，它们不仅对你阅读本书有益，而且还将帮助你开发自己的应用程序。要加入这些论坛，请遵循如下步骤：

(1) 访问p2p.wrox.com并点击Register（注册）；

(2) 阅读使用条款并点击Agree（同意）；

(3) 填写加入论坛所必需的信息以及你希望提供的一些可选信息，然后点击Submit（提交）；

(4) 将收到一封电子邮件，邮件内容描述了如何验证你的账号并完成加入论坛的流程。

不加入P2P，你也可以阅读论坛中的文章，但为了发表你自己的文章，你必须加入论坛。

在加入论坛之后，你就可以发表新的文章和回复别人发表的文章。你可以在任何时间通过Web阅读文章。如果你想要将某个特定论坛上新发表的文章通过E-mail发送给你，请点击论坛列表中该论坛名称前面的Subscribe（订阅）链接。

如果你想了解使用Wrox P2P的更多信息，请阅读P2P FAQ，它包含针对论坛软件工作情况以及许多针对P2P和Wrox图书的常见问题的解答。要阅读这份FAQ，请点击任意一个P2P页面上的FAQ链接。

致谢

值我生日之际，我坐在这里写这篇致谢。在去年的无数个漫漫长夜中，我伏案安排进度，制定计

划，偶尔甚至亲自完成一部分写作。当我承诺写作本书时，我并没有完全意识到完成这样一本书所需要的工作量和所需要克服的困难。我刚开始写作本书时还住在伦敦城外，在写作期间，我已决定离开英国，不到一年之后，我在位于美国马萨诸塞州的剑桥新家完成了本书。过去的一年不论对我个人而言，还是对我的职业而言，都发生了很多的变化，但在我的朋友和家人的支持下，我顺利地克服了这些困难。

首先，我要感谢Wiley出版社中和我一起工作的这个团队——Debra、Adaobi、Kit、Howard和Carol，以及许多其他参与本书出版工作的人员。我要特别感谢Kit Kemper，他忍受了我的写作时间表并使它正好在结束之前完成，而Debra Williams-Cauley从一开始就坚信这个写作计划是一个很好的主意。Howard Jones作为我的编辑帮助我保证内容的准确，出色地完成了他的工作。本书的诞生离不开在我的好朋友——我在共振仪器公司（后来的牛津仪器公司）的前任老板Malcolm Buckingham和Jamie McKendry的启发下产生的灵感，他们过去经常抱怨缺乏Linux专用的编程图书。同样地，如果没有来自我的几位好朋友——Kat、David Goodwin、Matthew Walton和Chris Aillon的贡献，本书也不会出现。谢谢你们，也谢谢Richard Blum，当我已显然不能按时完成本书时，你出现并加入了我们的团队，你做了大量的工作，我真的非常感谢你。

这一路上，还离不开来自我的幸福家庭的帮助——我的父母Paula和Charles、我的姐姐Hannah Wrigley和Holly、我的姐夫Joe，偶尔还有由我的祖母启发的灵感。我还受益于其他与我最要好的朋友，由于人数太多，这里就不一一列出了，但我要特别提到下面这些朋友：Hussein Jodiyawalla、Johannes Kling、Ben Swan、Paul Sladen、Markus Kobler、Tom Hawley、Sidarshan Guru Ratnavellu、Chris和Mad Ball（还有他的猫Zoe）、Emma Maule、John和Jan Buckman、Toby Jaffey和Sara、Sven Thorsten-Dietrich、Bill Weinberg、Daniel James、Joe Casad、Andrew Hutton和Emilie。我还要特别感谢我在Red Hat公司的所有朋友，我的老板和所有其他努力工作的同事，是你们使得我们的公司成为这个世界上最好的工作场所。Red Hat公司真正理解Linux开发的内涵，我非常感谢拥有这样一个优越的工作环境，它以真正的Linux社区精神鼓励参与这样一个项目——谢谢你们，你们太棒了！

最后，我要感谢来自Karin Worley的友谊，你为我提供了充分的机会在这个项目的最后阶段争取更多的时间。Karin，如果没有最近进入我的生活的这一新的幸福感，我不能确信是否能顺利地完成本书。

Jon Masters
剑桥，马萨诸塞

非常感谢Wiley出版社中为这个项目做出突出贡献的伟大团队。感谢组稿编辑Kit Kemper给我这个机会参与本书的写作。还要感谢加工编辑Howard Jones使一切步上正轨，并帮助使本书更加得体。我还要感谢Waterside产品公司的Carole McClendon为我安排了这一次机会，并一直在我的写作期间给予帮助。

最后，我要感谢我的父母Mike和Joyce Blum在我的成长过程中给予我的奉献和支持，感谢我的妻子Barbara和女儿Katie Jane、Jessica的爱、忍耐和理解，特别是当我在进行写作时。

Richard Blum

目　录

第1章 Linux 简介

为Linux编写软件的一个最大障碍是弄明白Linux是什么和它不是什么。不同的人对Linux有着不同的理解。虽然如今大多数用户都随意地将整个基于Linux的系统称为Linux，但从技术角度来说，Linux本身是由芬兰人Linus Torvalds编写的一个操作系统内核。在短短几年时间里，Linux迅速发展并被全球一些最大的企业和最强大的计算机用户所广泛接受。

Linux现在已成为一个提供高收益和企业质量的操作系统。它既用于一些最大型的超级计算机，也用在许多你根本不会想到的底层由Linux支持的最小型装置中。然而，这样一个在现代计算机领域中流行的大品牌却并不属于任何一家公司。Linux之所以会这么成功，是因为有数以千计来自世界各地的开发者在坚持不懈地努力来完善它。这些开发者和你一样，对编写高质量的软件有浓厚的兴趣，并从Linux社区中获取他人的经验。

不管Linux对你意味着什么，你选择本书是因为你想了解更多的有关如何成为一位专业Linux程序员的知识。当你准备开始这次学习之旅时，你将发现如果你对不同版本的Linux系统有所了解，知道如何开始对它们进行开发，并且清楚Linux开发和目前市场上其他流行平台的开发有何不同，将会对你的学习有很大的帮助。如果你已是一位Linux专家，那么只需略读本章即可。如果你想成为一位Linux专家，本章将会为你提供一些有用的指导。

在本章中，你将站在专业程序员的角度来学习什么是Linux以及Linux发行版的各个组件是如何组合在一起的。你将学习Linux系统上所用的大多数自由/开放源码软件（FLOSS）的开发过程，并找到大量提供开放源码革命动力的在线社区。最后，你还将了解到Linux与你之前遇到的其他操作系统的一些不同之处——我们将在本书的其余部分介绍更多这方面的内容。

1.1 Linux 发展简史

Linux有一个非常多样而有趣的历史，它的历史可能比你最初想象的要早得多。事实上，Linux的继承历史跨越了30年，可以从20世纪70年代最早的UNIX系统算起。这一事实不只是与执着的Linux狂热者有关，对读者来说也很重要，因为它至少让你对目前接触到的现代Linux系统的独特历史有一个大致的了解。通过介绍这些历史能让你更好地理解将Linux和目前市场上的其他操作系统区分开来的细节特征，并有助于使Linux的开发更加有趣。

Linux本身的工作最早始于1991年夏天，但早在Linux存在之前，就有了GNU项目。这个项目已花费了10多年的时间来创建很多必要的自由软件组件，其目的就是为了能创建一个完全自由的操作系统，如Linux。如果没有GNU项目，就不会诞生Linux；同样，如果没有Linux，你可能也不会立刻去阅

读GNU项目。这两个项目彼此之间互相受益，正如你将要在本书所要讨论的主题中发现的那样。

1.1.1　GNU 项目

1983年，那时的Richard Stallman（也称为RMS）还在麻省理工学院的人工智能实验室工作。直到那时为止，许多软件应用程序还是以源代码的形式提供，或有源代码可用，以便用户在必要的时候针对自己的系统进行修改。但从那时开始，已存在一种日益增长的趋势，即软件厂商只发行二进制版本的软件应用程序。软件的源代码很快变成了公司的“商业机密”，并受到高度保护——开放源码的开发者通常将这些源代码称为“秘笈”。

GNU项目最初的目标是通过使用必要的工具从源代码开始创建一个自由的类UNIX操作系统。该项目花了10多年的时间创建了所需的大多数工具，包括GCC编译器、GNU emacs文本编辑器和数十个其他的工具和文档。其中许多工具都以它们的高品质和丰富的功能而闻名，例如GCC和GNU调试器。

GNU享有许多早期的成就，但它在20世纪80年代缺少了一个最关键的组件。它没有自己的内核，即操作系统的核心，而是需要用户在已有的商业操作系统，如专有的UNIX上安装GNU工具。虽然这并不会对许多在他们自己的专用系统上使用GNU工具的用户造成什么影响，但如果GNU项目没有自己的内核，它就不是一个完整的项目。在Linux出现之前，针对是否开发这样一个内核（如发展中的GNU HURD）的激烈争论长期以来一直存在着。

Linux从来没有真正成为Richard Stallman所设想的GNU操作系统的一部分。事实上，尽管Linux已成为新一代用户和开发者的宠儿，并且是迄今为止更受欢迎的内核，GNU项目还是一直主张在其概念性的GNU系统中使用GNU HURD微内核。尽管如此，仍然会偶尔看到在提及一个完整的Linux系统时使用术语“GNU/Linux”，这是对在构建和运行任何一个现代Linux系统中扮演重要角色的众多GNU工具的认可。

1.1.2　Linux 内核

Linux内核的诞生远晚于GNU项目本身，在Richard Stallman发表他的最初宣言之后的10多年后它才出现。在此之前，已有一些可供选择的操作系统被开发出来，包括HURD微内核（在狂热的内核开发者社区之外只赢得了有限的大众关注）和由Andrew Tanenbaum编写的用于教学目的的Minix微内核。但由于种种原因，在Linux初次登台之前没有一个系统在这一段最好的时机中获得一般计算机用户的广泛认可。

就在那时，一位在赫尔辛基大学读书的年轻的芬兰学生正受制于Minix操作系统中很多他认为不合理的地方[①]。于是，他开始专门为他的（当时很高级的）AT 386微机设计自己的操作系统。此人就是Linus Torvalds，他将继续领导这个已创造了整个Linux产业的项目并激励着新一代。

Linus在1991年夏天开发出Linux的最初版本后，就在Usenet新闻组comp.os.minix中发表了如下的公告：

> 日期：25 Aug 91 20:57:08 GMT
> 组织：赫尔辛基大学

① 这些问题持续了数年之久，并成为早期Minix和Linux新闻组中大多数争论的主要话题。但在以后的几年中，争论渐渐平息，因为Linux已在市场上占据主导地位，而Minix及其后继者则继续为那些构思未来操作系统设计的人扮演学术研究的角色。

所有使用minix的人们——我正在为386（486）AT微机开发一个（免费的）操作系统（只是个人爱好，它不会像gnu那样庞大和专业）。我从4月份开始酝酿该系统，现在已进入准备阶段。我需要任何喜欢或不喜欢minix的朋友的反馈意见，因为我的操作系统与它有些类似（同样的文件系统物理布局（由于某些实际原因）以及其他方面）。

我已将bash（1.08）和gcc（1.40）移植到该系统中，并且它们看起来可以正常工作。这意味着在短短几个月内我就能够在该系统中做一些实际的事情，我很想知道大多数人都希望增加哪些功能。欢迎大家提出任何建议，但我无法保证一定会实现它们。

尽管Linus一开始很谦逊，但人们对Linux内核的兴趣却在全球迅速扩大。不久之后，Linux就推出了多个新版本，并且一个日益增长的用户群体（他们基本上都是开发人员，因为即使是简单地安装Linux也需要大量的专业技术）正在为Linux解决各种技术难题并将一些新的构思和想法付诸实现。现在的许多大名鼎鼎的Linux开发者都是在当时加入这一行业的。他们享受着能够在一个现代的完全免费的类UNIX系统上工作的乐趣，而不用忍受像其他系统那样的设计复杂性。

Linux的开发者利用许多已有的GNU工具来构建Linux内核并为它开发新的特性。事实上，在Linux出现后不久，就有越来越多的人对它发生了兴趣，Minix用户也开始切换到Linux系统上来工作——这些事情最终导致在Minix创造者Andrew Tanenbaum和Linus Torvalds之间发生了一系列著名的“口水战”。Tanenbaum至今仍坚持认为Linux的设计根本不如Minix。从理论上来说，这可能是事实，但对其他现代操作系统而言，可以说也存在着同样的问题。

读者可以从Peter H. Salus所著的*A Quarter Century of UNIX*（Addison-Wesley，1994）中了解到更多有关Linux和其他类UNIX操作系统历史继承的信息。

1.1.3 Linux发行版

随着Linux内核越来越受欢迎，人们希望Linux系统也能为那些对其内部编程机理没有深入了解的用户提供很好的服务。为了创建这样一个可用的Linux系统，需要的不仅仅只是一个Linux内核。事实上，如今一个普通的Linux桌面系统为了能提供从系统加电到具备丰富功能的图形桌面环境（如GNOME），它利用了成千上万个独立的软件程序。

当Linux第一次发布时，它还没有那么多丰富的软件可用。事实上，Linus开始时只有一个应用程序——GNU Borne Again SHell（bash）。那些曾经以受限的“单用户”模式（只运行一个bash shell）启动过Linux或UNIX系统的用户都知道这是一种什么体验。Linus使用一个单一的bash命令行shell对早期的Linux做了大量的测试，但是，即便这样一个单一的bash也不能直接运行在Linux系统上。它首先需要经过移植或修改才能从一个已有的系统（如Minix）转换到Linux系统上运行。

随着越来越多的人开始使用Linux并为Linux开发软件，那些有耐心编译和安装软件的用户已有大量的软件可用。但随着时间的流逝，人们发现以一种从无到有的方式来构建每一个Linux系统显然是一种不能忍受、不可复加的梦魇，除了最有热情的用户以外，它将阻止所有其他用户体验Linux所提供的功能。解决方法是使用Linux发行版的形式或将预先创建好的软件集以及Linux内核以软盘（或之后的光盘）形式提供给广泛的潜在用户群。

早期的Linux发行版只是简单地为那些不想自己从无到有构建整个系统的用户提供便利。它并没有跟踪系统上已安装了哪些软件或对软件的安全删除以及新软件的添加做出处理。直到包（package）