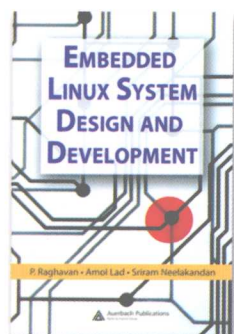


# 嵌入式Linux 系统设计与开发

Embedded Linux System Design and Development



P. Raghavan  
[印] Amol Lad 著  
Sriram Neelakandan

宋劲杉 王华勇 邱海霞 等译  
北京亚嵌教育研究中心 审校



嵌入式技术与应用丛书

# 嵌入式 Linux 系统 设计与开发

Embedded Linux System Design and Development

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书以嵌入式Linux为主题,详细解释了嵌入式Linux的系统设计模型。通过本书可以理解嵌入式Linux开发环境,为硬件平台创建Linux的板级支持包,理解Linux下的嵌入式存储;理解驱动编写和应用编写的模式,理解各种嵌入式Linux驱动,把应用程序从传统的实时操作系统移植到嵌入式Linux,在嵌入式Linux下编写实时应用程序,掌握在应用和驱动中寻找内存溢出和内存崩溃的方法,掌握剖析应用程序和内核的方法,理解uClinux的架构及其编程模型,理解嵌入式Linux的图形子系统。

本书可作为高等院校相关专业学生的教材,也可供软件工程师、测试和支持组人员参考。

Embedded Linux System Design and Development by P. Raghavan, Amol Lad and Sriram Neelakandan ISBN: 0-8493-4058-6 Copyright © 2006 by Auerbach.

Authorized translation from English language edition published by Auerbach, part of Taylor & Francis Group LLC; All rights reserved.

本书原版由Taylor & Francis出版集团旗下Auerbach出版公司出版,并经其授权翻译出版,版权所有,侵权必究。Publishing House of Electronics Industry is authorized to publish and distribute exclusively the Chinese language edition. This edition is authorized for sale throughout Mainland of China. No part of the publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版授权由电子工业出版社独家出版并限在中国大陆(不包括香港、澳门特别行政区及台湾)发行与销售,未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

Copies of the book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书封面贴有Taylor & Francis公司防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2007-1372

### 图书在版编目(CIP)数据

嵌入式Linux系统设计与开发/(印)(Raghavan, P.)等著;宋劲杉等译.-北京:电子工业出版社,2008.10  
(嵌入式技术与应用丛书)

书名原文:Embedded Linux System Design and Development

ISBN 978-7-121-07208-6

I. 嵌… II. ①拉… ②宋… III. Linux操作系统-程序设计-高等学校-教材 IV. TP316.89

中国版本图书馆CIP数据核字(2008)第117915号

责任编辑:王 纲

印 刷:北京东光印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×1092 1/16 印张:20 字数:563.2千字

印 次:2008年10月第1次印刷

定 价:48.00元

凡所购买电子工业出版社的图书有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系及邮购电话:(010)88254888。

质量投诉请发邮件至zlt@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010)88258888。

# 译者序

随着嵌入式系统的发展和软件复杂度的日益增加,出现了 VxWorks, Palm OS, Windows CE 等嵌入式操作系统,这些操作系统价格昂贵且源代码不公开,难以支持广泛的设备和应用软件。而 Linux 是一种免费的开放源代码的操作系统,使得基于 Linux 平台的开发和对各种设备的支持变得很容易,正迎合了嵌入式系统的需求,因此 Linux 近几年在嵌入式领域异军突起,成为最有潜力的嵌入式操作系统。许多 Linux 的改良版本,如 RTLinux (实时 Linux) 和 uClinux (用于无 MMU 设备的 Linux),进一步满足了特定应用场合的需求,使 Linux 成为应用领域最广泛的嵌入式操作系统。

本书是 Embedded Linux System Design and Development 一书的中文版。从书名就可以看出,该书着眼于设计 (Design) 和开发 (Development) 两个方面。作者具有非常丰富的嵌入式 Linux 系统设计经验和开发经验,并在书中系统地总结了这些经验,使本书具有极高的参考价值。具体来说,本书具有以下特点。

1. 本书为产品架构师提供了系统设计的指导方针。书中介绍了嵌入式 Linux 的各种解决方案和发行版,包括 uClinux 和 RTLinux,也介绍了各种可用的开发调试工具、性能分析工具、源代码、程序库和应用软件。此外,作者特别分析了 Linux 源代码的许可证问题,帮助读者合法地把开源软件应用到自己的产品中。
2. 本书将优化这一线索贯穿于各章节中。优化往往伴随着取舍,不同的产品应用对功能、性能和空间的要求不同,作者在书中随时列出各种可能的优化和取舍方案,并比较各种方案的优劣,供读者针对自己的产品应用来选择。
3. 本书详细讲解了移植 Linux 到嵌入式系统目标板需要解决的各方面问题,包括 BSP、Flash、设备驱动、应用程序和图形系统等方面,在各个关键的地方都附有源代码示例。本书汇集了各种关键问题的解决方法和注意事项,大大节省了读者在开发过程中摸索的时间。

本书不是一本入门书,不适合初学者,而是假定读者已具备 Linux 应用开发、驱动开发和嵌入式硬件架构的基础知识。本书重在经验的总结,尤其适合嵌入式 Linux 开发者和产品架构师。

本书的翻译与审校分工如下:前言由魏治宇翻译,第 1 章和第 2 章由黄一平翻译,第 3 章和第 9 章由刘意川翻译,第 4 章、第 5 章及所有附录由宋劲杉翻译,第 6 章和第 8 章由邸海霞翻译,第 7 章和第 10 章由王华勇翻译。全书由北京亚嵌教育研究中心 ([www.akaedu.org](http://www.akaedu.org)) 组织审校,感谢宋劲杉、邸海霞、魏治宇、何家胜和王华勇所做的审校工作。

本书的翻译工作得到了北京亚嵌教育研究中心、AKA 信息技术小组的郎铁山先生的大力支持。感谢电子工业出版社对我们的信任,把这样一项重要任务交给我们。最后,我们衷心感谢家人、师长、同事及朋友的鼎力支持。

由于译者水平有限,书中翻译不妥的地方恳请批评指正。

# 序

工业革命像一把利刃剑,把我们的生活由田园式的个体经营方式转变为分工合作的快节奏的现代生活方式。处身于这一时代的变革中,我们很难想象,今后的50年大多数人还会在松散、多变、无规则的嵌入式产品开发环境中开发,这是技术变革带来的明显转变。

我开发的第一个嵌入式软件项目没有使用现成的操作系统——因为没有,几年以后WindRiver公司才发明了VxWorks操作系统。在20世纪90年代中期,没有任何产品能够替代VxWorks的地位;但是,最近WindRiver公司宣布了一项基于Linux的产品。为什么会有这一改变?这是因为,现在绝大部分新产品的嵌入式操作系统是Linux。

我在一个开发HP LaserJet打印机固件的工程师小组工作了14年。这种打印机使用一个我们自己编写的操作系统,在我的记忆中称做LaserJet O.S.。通常,最优秀的工程师去做支持和扩展操作系统的工作,所有需要创建的LaserJet O.S.文档、测试用例都由工程师们一肩挑。这些天才的工程师们很少有时间和精力去关注竞争者们的产品的特性。我从这些经验和教训中学到的最重要的一课是:应该让大部分优秀工程师去完成赋予产品独一无二特性的工作,而架构方面应利用已有的成熟架构。对于那些需要复杂架构的产品来说,嵌入式Linux是一个很好的选择。

不管你是直接支持Linux还是为处理器购买Linux开发板支持包,都需要理解整个系统和某个特定子系统的细节。在本书中,作者做了大量工作,将所有嵌入式Linux开发必要的零碎知识综合在了一起。本书深入讨论了板级支持包、嵌入式存储和实时Linux编程,从一个嵌入式Linux开发者的角度,清晰地解释了嵌入式图形和uClinux。

在新产品的嵌入式操作系统中, Linux用户数量增长很快,这部分归因于嵌入式Linux可以非常方便地替代自己编写的操作系统。本书重点讲述在嵌入式系统上运行Linux,同时指导用户怎样从传统RTOS或自己编写的操作系统移植到嵌入式Linux。当需要TCP/IP网络、USB设备支持、数字安全支持及其他必要功能时,公司会考虑从现有的操作系统切换到Linux。不过,工程师们在未来的产品中使用并增强Linux功能主要是出于对Linux开发的乐趣。

在网络上有大量可用的Linux信息,我认为Linux是到目前为止文档最多的软件。那么怎样在有这么多有用资料的前提下编写一本关于嵌入式Linux的有价值的书呢?首先,嵌入式Linux及相关软件的应用范围如此庞大,那么寻找真正有用的内容是一个很大的挑战。理解Linux上的各项目是如何独立运作和协同工作的可以帮助你深入理解嵌入式Linux的产业生态系统。其次,获取正确信息是个技术活。在一个嵌入式设备中,为使系统工作正常,需要开发引导程序、内核和包含应用程序的文件系统,深入理解这几个模块之间的依赖关系和构建这3个映像的开发运行环境并不是一件容易的事情。另外,当你遇到一个问题的时候,深入理解可用的调试工具和嵌入式设备的调试技术可以节省大量的时间和精力。

最后,阅读本书的最佳理由是:这项技术非常有吸引力。如果你用老式方法开发过一个仅有单个可执行映像的嵌入式产品,那你必然会被嵌入式Linux的易用和强大迷倒。对于嵌入式开发新手而言,他们的嵌入式开发环境可以和桌面PC一样强大和容易工作。

Todd Fischer  
Cadenux 总裁和创始人



# 前 言

20世纪90年代中期,我们还在大学的时候,就听说了一项令人兴奋的新技术——因特网,它对我们的生活造成了强烈的冲击。随着因特网的普及,我们又听说了一个开源的操作系统——Linux,它由全世界数千名程序员共同开发而成。Linux给了我们一次深入理解操作系统内核的机会,很快我们就变成了Linux的热衷者。我们发现Linux不仅仅是一个操作系统,它还是人类历史并行发展的产物,它基于人类的尊严、选择和自由的概念,像我这样的年轻程序员可以通过Linux掌握最新的技术。

当我们成为嵌入式专家时, Linux在嵌入式市场上也展现了它的强壮特性。尽管如此,我们也看到了一些新的运动:在Linux内核上运行一个硬实时的内核;在没有存储管理单元的微控制器上运行Linux。当客户请求我们在基于MIPS的SoC上,将我们的软件从商业实时操作系统移植到嵌入式Linux之上的时候,我们掩饰不住内心的喜悦。经验告诉我们,通向嵌入式Linux的道路不是那么平坦的,原因主要有以下几点。

1. 毋庸置疑,因特网上有很多关于嵌入式Linux的信息,但是这些信息过于分散,不容易形成一个统一的模式,因此,将这些信息转化成知识基点是一项很耗时的任务。而许多以产品为基本业务的公司缺的就是时间,它们需要快速决定需求并迅速执行。那么,一项错误的决定,尤其是在版权这样的关键问题上的决定,会给公司带来毁灭性的灾难。
2. 有一个非常错误的概念很流行,那就是嵌入式系统都是关于硬件或操作系统的。计算能力随着摩尔定律快速地增长,进入嵌入式系统的应用软件的数量也以同样的速率增长,因此,应用程序已经变为嵌入式系统用户空间程序。这样,构建一个基于Linux的嵌入式系统并没有止步于操作系统,在编写和构建应用的时候也有很多事情要做。同时,应用程序有它自己的一套不同于操作系统的问题,如版权、工具链等。
3. 嵌入式Linux和商业的实时操作系统不同。商业操作系统仅提供单一的支持点,如补丁和文档。嵌入式Linux采用了一个新的开发策略。通常,开发者需要从各种各样的邮件列表中寻找补丁的相关信息,而这又是很耗费时间的。

当我们最终在嵌入式Linux上成功实现了各种应用的时候,我们决定向世界分享我们的部分思想和经验,这些思想和经验的结晶就是这本书。本书包含了整个嵌入式Linux系统开发的路线图。我们的首要目标是让读者意识到嵌入式Linux开发过程中会出现各种各样的问题。

本书的主题有两个:

- 从传统实时操作系统中快捷地切换到嵌入式Linux。
- 解释嵌入式Linux的系统设计模型。

## 读者可从如下方面获益

本书提供了一些方法,这些方法可解决开发人员面对嵌入式Linux环境编程时遇到的问题。下面列举了一些普遍存在的问题。

- 如何理解嵌入式 Linux 开发模型。
- 怎样在嵌入式 Linux 下编写、调试、剖析应用和驱动。
- 怎样弄懂嵌入式 Linux 的板级支持包的架构。

本书提供了上述问题的实际解决方案。

读完本书后，读者可以：

- 理解嵌入式 Linux 开发环境。
- 理解并为硬件平台创建 Linux 的板级支持包。
- 理解 Linux 下的嵌入式存储、驱动编写和应用编写的模式。
- 理解各种嵌入式 Linux 驱动，如串口、I2C 等。
- 把应用程序从传统的实时操作系统移植到嵌入式 Linux。
- 掌握在嵌入式 Linux 下编写实时应用程序。
- 掌握在应用和驱动中寻找内存溢出和内存崩溃的方法。
- 掌握剖析应用程序和内核的方法。
- 理解 uClinux 的架构和其编程模型。
- 理解嵌入式 Linux 的图形子系统。

同时，本书适用于管理者，可以帮助他们对嵌入式 Linux 的发布版本进行选择，为转移到嵌入式 Linux 制定路线图，以及在商业产品中使用 Linux 的版权模型。

## 读者对象

### 首要的读者

- **架构师**：他们更多地关注于实时性问题、性能问题和移植计划。
- **软件工程师**：他们需要获得这项技术方方面面的细节。

### 次要的读者

- **法务人员**：因为绝大多数嵌入式产品有它自身的所属权，如果对版权理解错误，会对公司造成不利的局面。
- **经理**：他们需要考虑发行版、版本、工具集和厂商的选择问题。
- **测试和支持组**：测试和支持队伍需要进行培训，因为当产品切换到嵌入式 Linux 的时候，从使用习惯上会有一些变化。

## 背景知识

希望读者能够具有任何一种嵌入式操作系统的基本知识和编程经验。本书不是一本 Linux 内核的书籍，只会出现一些基本的 Linux 内核概念和用户空间编程模型。

本书努力做到和内核的版本无关；不过，在必要的地方，我们会使用 2.4 版或 2.6 版的内核作为示例。

## 源代码的获得

读者可以在如下网址获取源代码：[http://www.crcpress.com/e\\_products/downloads/download.asp?cat\\_no=AU0586](http://www.crcpress.com/e_products/downloads/download.asp?cat_no=AU0586)

# 概 述

本文由 10 章和两个附录组成。

第 1 章简单介绍了嵌入式 Linux 的历史,说明了嵌入式 Linux 和其他实时操作系统相比的优势,并详细讨论了各种开源和商业的嵌入式 Linux 可用发行版的特性。本章总结了从传统的实时操作系统到嵌入式 Linux 转换的路线图。

第 2 章解释了嵌入式 Linux 的体系结构,并与其他传统实时操作系统和内核架构进行了对比,介绍了各种 Linux 内核子系统(如硬件抽象层、内存管理、调度器和文件系统等),给出了一个用户空间 Linux 编程模型的小型描述。本章的第二部分描述了 Linux 的启动顺序,从引导程序到内核的启动和用户空间的启动脚本。最后一部分描述了构建 GNU 交叉平台工具链的步骤。

第 3 章描述了引导程序的架构,讨论了系统的内存映射,包括硬件和软件的内存映射。第二部分描述了中断管理、PCI 子系统、定时器、UART 和电源管理的细节。

第 4 章描述了可访问闪存设备的 MTD 子系统的架构。本章第二部分讨论了各种嵌入式文件系统(如 RAMFS, CRAMFS, JFFS2 和 NFS 等),讨论了嵌入式系统的各种存储空间优化方法(包括内核空间和用户空间的优化),还讨论了各种为嵌入式 Linux 设计的应用程序(如 Busybox)。本章最后给出了优化内核内存空间的步骤。

第 5 章详细讨论了嵌入式驱动,如串口驱动、以太网驱动、I2C 子系统和 USB Gadget。

第 6 章讨论了把应用程序从传统的实时操作系统移植到嵌入式 Linux 的路线图。本章的其余部分描述这个移植过程的细节,首先讨论了 Linux 线程,然后讨论了操作系统移植层,最后讨论了内核的 API 驱动。

第 7 章讨论了 Linux 的实时特性,其中描述了各种内核中涉及的延迟(如中断延迟和调度延迟),讨论了改善内核响应时间的方案(如内核抢占和 O(1)调度)。本章的核心是讨论 Linux 的 POSIX.1b 编程接口。本章详细描述了各种 POSIX.1b 实时扩展(如实时调度、内存锁定、消息队列、信号量和异步 I/O)。最后一部分简要描述了 Linux 实现硬实时的途径,即 RTAI 的实时编程模型。

第 8 章分成了三个部分:编译、调试和剖析。第一部分描述了各种编译内核和用户空间应用程序的机制。第二部分描述了如何用一些工具(如 mtrace, dmalloc 和 valgrind)来调试内存问题。最后一部分讨论了用 eProf, OProfile 和增强内核函数的剖析方法来剖析用户空间和内核函数。

第 9 章详细描述了通用帧缓冲区和怎样使用帧缓冲接口编写应用程序,同时简要讨论了 X 图形子系统,以及它为什么不适用于嵌入式设备。最后一部分描述了 Nano-X 窗口环境。

第 10 章描述了 uClinux 的架构和编程环境。第一部分描述了基于 uClinux 系统怎样加载和运行程序,并讨论了 uClinux 的内存管理、进程创建和共享库。第二部分描述了 XIP 和怎样把应用程序从标准 Linux 移植到 uClinux,同时描述了怎样在 uClinux 下编译应用程序。

附录 A 描述了减少 Linux 启动时间的各种技术。

附录 B 讨论了 GPL 对于嵌入式 Linux 的含义,以及私有软件如何在嵌入式 Linux 下保证安全。可用的源代码可以从下面的网址下载:

[http://www.crcpress.com/e\\_products/downloads/download.asp?cat\\_no=AU0586](http://www.crcpress.com/e_products/downloads/download.asp?cat_no=AU0586)



# 目 录

第 1 章 引言 .....	1
1.1 嵌入式 Linux 的发展史 .....	2
1.2 为什么选择嵌入式 Linux .....	5
1.3 嵌入式 Linux 与桌面 Linux 之比较 .....	7
1.4 常见问题 .....	8
1.5 嵌入式 Linux 发行版 .....	10
1.6 移植方案 .....	19
第 2 章 Linux 简介 .....	21
2.1 嵌入式 Linux 的体系结构 .....	21
2.2 Linux 内核结构 .....	23
2.3 用户空间 .....	27
2.4 Linux 启动过程 .....	30
2.5 GNU 交叉编译工具链 .....	35
第 3 章 板级支持包 .....	42
3.1 编译内核过程中与 BSP 有关的工作 .....	43
3.2 boot loader 接口 .....	44
3.3 内存映射 .....	47
3.4 中断管理 .....	52
3.5 PCI 子系统 .....	56
3.6 定时器 .....	59
3.7 串口 .....	59
3.8 电源管理 .....	60
第 4 章 嵌入式存储系统 .....	64
4.1 Flash 映射 .....	64
4.2 MTD —— 存储技术设备 .....	65
4.3 MTD 架构 .....	67
4.4 NOR Flash 的 MTD 驱动程序示例 .....	69
4.5 Flash 映射驱动程序 .....	79
4.6 MTD 字符设备和块设备 .....	85
4.7 Mtdutils 软件包 .....	86
4.8 嵌入式文件系统 .....	87
4.9 优化存储空间 .....	89
4.10 内核的运行时内存调优 .....	93

<b>第 5 章 嵌入式设备驱动</b> .....	96
5.1 Linux 串口驱动程序 .....	97
5.2 以太网驱动程序 .....	106
5.3 Linux 中的 I2C 子系统 .....	111
5.4 USB Gadget .....	117
5.5 看门狗定时器 .....	124
5.6 内核模块 .....	125
<b>第 6 章 应用程序移植</b> .....	128
6.1 体系结构的对比 .....	128
6.2 应用程序移植路线图 .....	129
6.3 使用 pthread 编程 .....	132
6.4 操作系统移植层 .....	141
6.5 内核 API 驱动 .....	149
<b>第 7 章 实时 Linux</b> .....	157
7.1 实时操作系统 .....	157
7.2 Linux 和实时性 .....	158
7.3 Linux 中的实时编程 .....	163
7.4 硬实时 Linux .....	198
<b>第 8 章 编译和调试</b> .....	206
8.1 编译内核 .....	207
8.2 编译应用程序 .....	213
8.3 编译根文件系统 .....	216
8.4 集成开发环境 .....	218
8.5 调试虚拟内存问题 .....	220
8.6 内核调试器 .....	229
8.7 性能评价 .....	231
<b>第 9 章 嵌入式系统中的图形子系统</b> .....	243
9.1 图形系统 .....	243
9.2 Linux 桌面图形——X 图形系统 .....	244
9.3 显示硬件介绍 .....	246
9.4 嵌入式 Linux 系统中的图形 .....	248
9.5 嵌入式 Linux 图形驱动程序 .....	248
9.6 窗口环境, 工具包和应用程序 .....	264
9.7 小结 .....	268
<b>第 10 章 uClinux</b> .....	269
10.1 无 MMU 系统上的 Linux .....	269
10.2 程序装载和执行 .....	271
10.3 内存管理 .....	283

10.4	文件/内存映射——uClinux 中 mmap()函数的复杂性 .....	287
10.5	进程创建 .....	288
10.6	共享库 .....	289
10.7	往 uClinux 上移植应用程序 .....	292
10.8	XIP 片内执行 .....	298
10.9	编译 uClinux 发行版 .....	298
附录 A	快速启动 .....	301
附录 B	GPL 与嵌入式 Linux .....	304

# 第1章 引言

嵌入式系统是为了完成一组特定的任务而设计的专门的计算机系统。嵌入式系统的历史可以追溯到20世纪60年代末,当时嵌入式系统被用来控制机电式的电话交换机。最早被认可的嵌入式系统是由 Charles Draper 和他的小组开发出来的阿波罗导航计算机。随后,嵌入式系统被应用于军事、医药科学、航空及汽车工业。现在,嵌入式系统被广泛应用于各个领域。例如:

- 网络设备,如防火墙、路由器和交换机等。
- 消费电子,如MP3播放器、手机、PDA、数码相机和家庭娱乐系统等。
- 家用电器,如微波炉、洗衣机、电视等。
- 关键任务,如卫星和飞机控制器等。

以下列出了嵌入式系统与桌面电脑的主要不同之处:

- 嵌入式系统通常对价格敏感。
- 大多数嵌入式系统有实时要求。
- 有多种体系结构的CPU可被用于嵌入式系统(如ARM, MIPS和PowerPC等)。嵌入式系统根据目标应用选取专用的处理器。举例来说,数码相机里的处理器都是为捕捉和传递影像而定制的。
- 与桌面系统相比,嵌入式系统在RAM、ROM和其他I/O设备方面资源稀缺。
- 对于大部分嵌入式系统来说,电源管理是一个重要的问题。
- 嵌入式系统的开发调试环境与桌面系统有很大不同,嵌入式系统通常有一个用于调试的内置电路。
- 嵌入式系统是为了某一种或一组专门的应用,从硬件和软件两个角度来设计的。例如,MP3播放器里有一个内置的硬件MP3解码器。

实际上,早期的嵌入式系统并没有通用的操作系统。所有底层软件都是由各公司自己开发的,这些软件直接驱动硬件,几乎没有或只有极少量的多任务处理和用户交互功能。但是,随着时间的推移,更加复杂的嵌入式系统开始出现,这些嵌入式系统需要越来越多的共同特性。因此,在嵌入式系统中使用一种操作系统是十分必要的,嵌入式操作系统至少需要提供多任务/多线程、进程与内存管理、进程间通信和定时器等。因此很多公司开始加强他们自己开发的底层软件,以便在他们的嵌入式系统中运行一个微型的但是功能齐全的操作系统。众多公司开始致力于提供针对嵌入式系统的操作系统。

如今有大量的嵌入式操作系统。除了各公司自己开发和使用的操作系统,还有一些通用的嵌入式操作系统,例如Wind River的VxWorks、Microsoft Windows CE、QNX Neutrino、Accelerated Technology的Nucleus、Red Hat的eCos、Sun Microsystems的ChorusOS、LynuxWorks的LynxOS和嵌入式Linux等。

## 1.1 嵌入式 Linux 的发展史

1991年，赫尔辛基大学的 Linus Benedict Torvalds 始创了 Linux 操作系统。他发表在 Minix 开发邮件列表里的信（见清单 1.1）被认为是 Linux 革命的开始。

从那以后 Linux 就一直不断地向前发展。Linux 的开源开发模型和基于 GNU 公共许可证 (GPL) 发布的特征，吸引了世界范围内成千上万的开发人员为之奉献。GPL 许可证使 Linux 内核的所有源代码都可以被自由免费地用于个人或商业目的。由于 Linux 内核源代码可以自由获得，很多开发人员为 Linux 内核贡献代码。来自全球开发人员的支持使 Linux 成为一个高度可靠、鲁棒、功能强大的操作系统。1996年初，作为 Michael Barabanov 和 Victor Yodaiken 的一个研究项目，Linux 开始被应用于硬实时嵌入式系统。这个 RT-Linux 研究项目基于一个小型的源于 Linux 的实时内核提供硬实时的时限保证。1997年，旨在将 Linux 运用于无 MMU 的处理器器的 uClinux 项目启动，并于 1998 年正式发布。1999年至 2004年，Linux 被广泛运用于嵌入式系统，下面几节将对这一阶段嵌入式 Linux 系统的主要发展进行介绍。

### 清单 1.1 Linux 的起源

---

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and Professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

### 1.1.1 1999 年

1999 年是 Linux 开始在嵌入式系统领域打下基础的一年。其中一些重大的发展如下。

- 1999 年 9 月的嵌入式系统大会 (ESC) 上，包括 Lineo, FSM Labs, MontaVista 和 Zentropix 在内的一些公司宣布对嵌入式 Linux 提供支持。



- Zentropix 建立了 RealTimeLinux.org 用于讨论实时 Linux 解决方案的可能性。
- Lineo 宣布成立 Embedded Advisory Board (EMLAB), 讨论将 Linux 用于嵌入式领域的可能性。
- Rick Lehrbaum 开启了一个嵌入式 Linux 的门户: Linuxdevices.com。
- Paolo Mantegazza 发布了 RTAI, 为 Linux 增加了硬实时方面的支持。
- Lynx 实时系统 (如今的 Lynux-Works) 发布了 BlueCat Linux, 成为嵌入式 Linux 的第一个商业发行版本。

### 1.1.2 2000 年

2000 年, 嵌入式 Linux 开始被许多公司运用到它们的生产线上。

- Samsung 推出 Yopy, 这是一种采用 Linux 的 PDA。
- Ericsson 推出 HS210, 这是一种基于 Linux 的无线可视电话, 将无线连接与因特网、电话、电子邮件功能结合起来。
- Atmel 推出一种基于 Linux 的因特网设备——AT75C310, 支持 VoIP 和音频。
- Agenda Computing 在 Linuxworld 大会上展示了一种基于 Linux 的 PDA。

在这一年, Linux 的实时支持也受到越来越多的关注。

- TimeSys 公司推出 Linux/RT, 它是一种通过资源预留技术提供可预测的应用程序响应时间的嵌入式 Linux。
- MontaVista Software 启动了 Linux Real-Time Characterization Project, 为开发人员提供了一套用于测量 Linux 系统的实时响应的开源工具。
- Red Hat 发布了为 Linux 增加实时支持的 EL/IX 规范 1.1 版。

这一年, 很多用于嵌入式 Linux 的工具和应用程序相继推出。

- Busybox 0.43 发布。它是 Busybox 推出的第一个稳定版本。
- GoAhead Software 发布了用于嵌入式 Linux 应用程序的 GoAhead Web 服务器。
- Trolltech 推出了 Qt/Embedded, 一个针对嵌入式 Linux 的 GUI 应用程序框架和视窗系统。
- Greg Haerr 公司推出了 ViewML 嵌入式浏览器, 它基于 Microwindows 视窗系统。

在同一年, Rick Lehrbaum 成立了 Embedded Linux Consortium (ELC), 其成员包括 Intel 和 IBM 等大公司。这个联合机构的目的是使 Linux 和开源软件在嵌入领域的应用更加方便。ELC 致力于将 Linux 发展成为高效、安全、可靠的嵌入式操作系统。

HP, Intel, IBM 和 NEC 成立了以提供企业 Linux 解决方案为目标的 OSDL (Open Source Development Lab)。

### 1.1.3 2001 年

2001 年最大的成果是 Linux kernel 2.4 的问世, 它随后被用于许多嵌入式 Linux 发行版。在这一年, Linux 被广泛用于各种手持设备和小型设备。

- Sharp 电子推出了基于 Linux 的 PDA。
- Trolltech 和 Lisa 为 Compaq iPAQ 掌上电脑开发出一种无线 iPAQ 解决方案。

- NeoMagic 也推出了一种用于智能手持设备的基于 Linux 的 SoC 平台。
- Transmeta 公司推出“Midori” Linux，是面向小型设备的 Linux 发行版。

嵌入式 Linux 的标准化也在 2001 年取得了进展。

- 包括 Toshiba 和 NEC 在内的大公司联合成立了日本嵌入式 Linux 联盟 (EMBLIX)，旨在推动嵌入式 Linux 在日本的教育和标准化。
- TV Linux 联盟成立，其主旨是定义一套在机顶盒上使用 Linux 的标准。Broadcom, Motorola 和 SUN Microsystems 都参与了这个组织的建立。
- 自由标准组织 (FSG) 推出了 Linux Standard Base (LSB) 1.0 规范。LSB 的目标是建立一套标准以加强不同的 Linux 发行版之间的兼容性，从而使应用程序能够在任何一个兼容的 Linux 系统上运行。LSB 得到了业内公司的广泛认可，并被认为对嵌入式 Linux 同样有价值。

在工具和应用程序方面也取得了如下一些进步。

- uClibc 的第一个主版本 uClibc 0.9.8 首次亮相。uClibc 现在已成为几乎所有的嵌入式 Linux 发行版不可或缺的部分。
- IBM, SuSE, Red Hat, QNX Software Systems, Borland 和 Merant 联合成立了 Eclipse 联盟，其目的是为嵌入式系统提供一个开发环境框架。现在包括 TimeSys, LynuxWorks, MontaVista 在内的许多公司都在使用 Eclipse 框架为嵌入式 Linux 开发提供 IDE。

#### 1.1.4 2002 年

2002 年是 Linux 在嵌入系统市场上取得重大发展的一年，越来越多的公司在它们的产品设计中采用 Linux。Linux 的实时支持也进一步完善。

- Robert Love 的占先式内核补丁、Andrew Morton 的低延迟补丁及 Ingo Molnar 的 O(1) 调度器被应用到 Linux 内核。
- RTLinux 为用户空间程序提供了硬实时的能力。
- ADEOS 项目推出了 ADEOS 的第一个发行版，该系统通过一个硬件抽象层使实时内核和通用操作系统并存。

在标准化方面的努力取得了如下进展。

- ELC 推出了嵌入式 Linux 联盟平台标准 (ELCPS)。ELCPS 为 API 层提供了一种标准以提高程序代码的复用性和可移植性。这种标准可以帮助开发人员降低嵌入式 Linux 应用程序开发的时间和成本。
- OSDL 成立了电信级 Linux (CGL) 工作小组，他们的任务是促进并标准化 Linux 在电信级系统中的应用。同年，CGL 推出了 CGL 需求规范 1.x 版。
- 自由标准组织推出了 LSB 1.1 和 LSB 认证项目，这个认证项目的目的是通过独立的权威机构认证某个 Linux 发行版或者应用程序是否符合 LSB。

同年，Linux 在数码娱乐产业中得到了更为广泛的应用。Intel 推出一款家用数码媒体适配器的参考设计。Trace Strategies Inc. 发表了一份研究报告，认为 Linux 是诸如数字交互式电视 (ITV)、机顶盒等设备上的首选操作系统。

也在这一年，uClinux 从 SnapGear 和 Ridgerun 那里得到了共享库的支持。随后进入了主流 Linux 内核 2.5.46 版。

### 1.1.5 2003年

2003年, Linux在手机和小型办公室/家庭办公室(SOHO)市场上取得了进展。

- 摩托罗拉宣布其 A760 手机使用 Linux 作为嵌入式操作系统。
- Linux 逐渐被应用于 SOHO 和消费市场上的网关、路由器和无线局域网中。

这一年, 在标准化方面做出了更大的努力。

- ELC 扩展了 ELCPS, 以增加对电源管理、用户界面和实时标准的支持。
- OSDL 发布了 CGL v2.0, 它在安全性、高可用性和集群方面有重大进步。
- 消费电子 Linux 论坛 (CELF) 于 2003 年 6 月成立。它为 Linux 在消费电子设备上的应用提供规范, 并且维护了一套专门为消费电子设备上的应用增强的 Linux 内核代码树。CELF 号召各大公司为这个代码树贡献代码, 使得 Linux 成为消费电子设备操作系统的事实标准。Matsushita, Sony, Hitachi, NEC, Royal Philips 电子, Samsung, Sharp 和 Toshiba 公司等是 CELF 的发起者。

2003 年末 Linux 2.6.0 内核发布。

### 1.1.6 2004年

2004 年发生的一些重大事件如下。

- LynuxWorks 发布了基于 2.6 版 Linux 内核的 BlueCat Linux 发行版。这是第一个基于 2.6 版 Linux 内核的商业嵌入式 Linux 系统。
- Sony 公司推出了基于 Linux 的日本车内导航设备和信息娱乐系统, 这些设备具有 3D 地图导航技术、媒体播放器、硬盘驱动器、GPS 和 PC 接口。
- Trolltech 宣布推出一种智能手机上的 PDA 应用开发包。
- OSDL 的 CGL 规范被电信业界广泛接受。
- CELF 推出了它的第一个关于在消费电子设备上使用 Linux 的规范。同时还以内核源码树补丁的形式提供了 9 种目标版的参考实现。
- 自由标准组织 (FSG) 和 OSDL 发布了 LSB 2.0。

现在, 很多公司都将嵌入式 Linux 用于它们的新产品设计。越来越多的厂商正在提供各种硬件平台上的嵌入式 Linux 发行版。如今, 嵌入式 Linux 已成为运行嵌入式系统的首选操作系统。芯片提供商 AMD, ARM, TI, Motorola, IBM, Intel 等都纷纷选择 Linux 作为硬件开发的平台。Sony 和 NEC 等消费电子设备厂商将 Linux 用于 DVD, DVR 和手机。

## 1.2 为什么选择嵌入式 Linux

任何一个初涉嵌入式 Linux 的人都会被这样一个问题所困扰: “为什么选择嵌入式 Linux 作为目标平台上的操作系统?” 这一节我们将讨论与私有知识产权的嵌入式操作系统相比嵌入式 Linux 的优势所在。

### 1.2.1 厂商中立

如果选择私有知识产权的操作系统, 那么你产品的整个生命周期都被锁定在这个操作系统的提供商了。如果提供商不能提供很好的服务支持将会导致产品上市时间延迟。为了解决一个小小的问

题,你也许需要等上数天甚至数周的时间。然而,更换操作系统提供商又意味着产品的生命周期要重新来过。

嵌入式 Linux 是厂商中立的,所有嵌入式 Linux 产品的提供商都有基本相同的业务模型。产品也都是同一内容的不同变体,它们都有相同的基本组件,如 Linux 内核、程序库、基本命令工具等。如果你觉得你的嵌入式 Linux 发行版的提供商不能满足产品要求的话,你可以以相对较低的成本更换提供商。由于 Linux 内核源代码和相关工具很容易获得,你也可以不依靠提供商而自己开发。

## 1.2.2 产品上市时间

嵌入式 Linux 拥有丰富的开发工具和应用程序。大多提供商为各种硬件平台提供试用版工具,这些工具都可以免费下载。很有可能已经有针对你的硬件平台的可用的 Linux 移植了。因此你只需要把精力放在应用程序开发上,而无需操心如何将 Linux 移植到你的硬件平台上,也无需操心如何为你的高性能 I/O 卡编写设备驱动程序。采用嵌入式 Linux 系统,产品可以快速出炉。

在嵌入式平台上采用 Linux 系统的一个好处是节约了开发时间。如果采用基于 Linux 的主机开发环境,大多数需要在目标硬件上运行的应用程序都可以在 Linux 主机上测试,减少了移植应用程序的时间。例如,如果你的目标平台需要一个 DHCP 客户端,你可以获取任何开源的 DHCP 客户端程序(这个程序的大小要适合目标平台的存储要求,不能太大),在一个 Linux 主机上编译测试。如果它能在 Linux 主机上运行,那么仅仅需要将其针对目标平台交叉编译,它应该也能在目标系统上正常运行。

## 1.2.3 硬件支持的多样性

对于私有知识产权的嵌入式操作系统的开发商来说,目前新的高性能、低成本和更为精巧的微处理器和 I/O 设备大量出现,支持它们变得越来越困难。高端的硬件设备已经有了,但用户却很可能无法使用它,因为私有知识产权的嵌入式操作系统提供商无法提供支持。

Linux 能够支持众多的系统结构和高端 I/O 设备,因此你能够自由地选择使用适合你系统的硬件。Linux 是任何硬件或软件创新研究的首选平台。在大学里,它作为研究和学习的工具而得到了广泛使用。Linux 还是硬件制造商最受青睐的开发平台。

## 1.2.4 低成本

嵌入式 Linux 能以最低成本满足开发、培训和雇佣员工的要求。

### 开发成本

私有知识产权软件的提供商要花费大量资金用于购买开发工具的许可证。这些许可证一般都是每客户许可(Per Seat)的,因此限制了使用这个开发环境的用户人数。如果使用嵌入式 Linux,组成开发环境的所有开发工具和应用程序,如编译器、链接器、程序库和 Shell 等都可以免费下载。用户还能花费很少的钱甚至免费获得优秀的 IDE。此外,还能获得基于 GUI 的配置环境和性能分析工具。

### 培训和雇用成本

新的开发环境相当昂贵。如果你需要重新培训开发人员,或者你决定雇用熟悉私有知识产权操作系统的开发过程、API 用法、优化技术的专家,那么产品的生产成本就会显著增加。Linux 采用基于 UNIX 的编程模型,大多数工程师对它都很熟悉。因此,采用嵌入式 Linux 的培训成本很低。