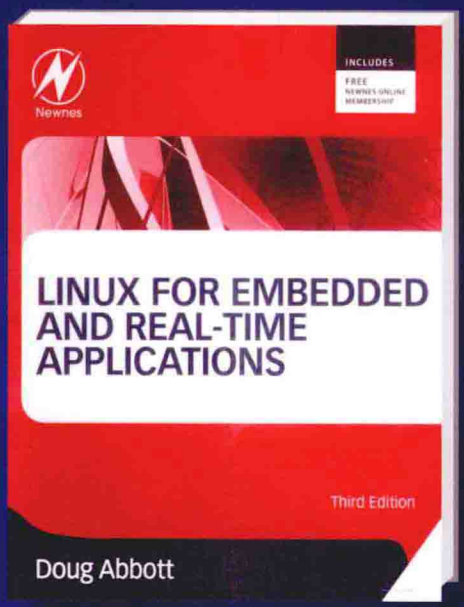
HZ BOOKS
华章科技

ELSEVIER
爱思唯尔

这是一本实用、全面、系统的嵌入式Linux入门书籍。
围绕实时应用逐章分模块，从基本概念、原理、配置到实战案例全面解构嵌入式Linux。
以ARM单板机为平台，涵盖Linux基础、交叉编译、Linux组件和工具三大部分内容。
本书不拘泥于形式，立足实战，为读者提供详细的操作指南。

电子与嵌入式系统
设计译丛



Linux for Embedded and
Real-Time Applications
Third Edition

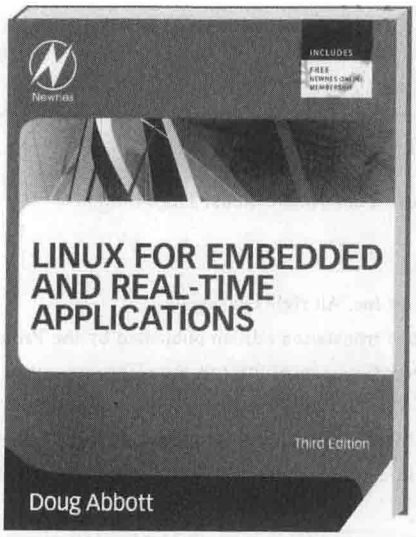
Linux嵌入式实时应用开发实战

（原书第3版）

［美］Doug Abbott 著　周艳 译

机械工业出版社
China Machine Press

电子与嵌入式系统
设计译丛



Linux for Embedded and
Real-Time Applications
Third Edition

Linux嵌入式实时应用开发实战

（原书第3版）

［美］Doug Abbott 著　周艳 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Linux 嵌入式实时应用开发实战（原书第 3 版）/（美）阿伯特（Abbott, D.）著；周艳译．
—北京：机械工业出版社，2015.1
（电子与嵌入式系统设计译丛）
书名原文：Linux for Embedded and Real-Time Applications, Third Edition

ISBN 978-7-111-48857-6

I. L… II. ① 阿… ② 周… III. Linux 操作系统 IV. TP316

中国版本图书馆 CIP 数据核字（2014）第 295627 号

本书版权登记号：图字：01-2013-6484

本书介绍目前广泛应用于嵌入式产品的 Linux 系统开发，包括 Linux 系统特性、环境配置、交叉开发环境中的应用编程，以及 Linux 开发组件和工具，并辅以相关参考资料，对于初次在嵌入式和实时领域应用 Linux 的工程技术人员来说，是一本十分详尽的指导书。

Linux 嵌入式实时应用开发实战（原书第 3 版）

出版发行：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：张国强　　责任校对：殷　虹
印　　刷：三河市宏图印务有限公司　　版　　次：2015 年 1 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：14
书　　号：ISBN 978-7-111-48857-6　　定　　价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

译 者 序

Linux 系统是开源代码，便于移植和裁剪，在一个 2MB 的 Flash 存储器上便可以运行功能完整的 Linux 程序。Linux 在应用 32 位以及 64 位处理器的嵌入式产品中得到了很广泛的应用，如手机、多媒体设备、PDA 等。正是基于这样的背景，作者写作了本书。对于初次在嵌入式和实时领域应用 Linux 的工程技术人员来说，这是一本详尽的指导书。

本书第一部分以 Fedora 为例，图解在 Windows 中安装和配置 Linux 工作站的方法和步骤，着重介绍与嵌入式开发者相关的 Linux 特性和主机开发环境的配置，以 Friendly ARM 公司生产的 Mini2440 开发板为目标板，介绍其资源和配置，并简单介绍了集成开发环境 Eclipse 平台的安装和使用。

第二部分介绍了交叉开发环境中的应用编程。首先以一个程序为例，介绍如何从用户空间访问硬件，如何在目标板上使用 GDB 来调试代码；接着，在 Linux 多任务的系统里，介绍如何应用 Posix 线程实现命令协议交互，如何在嵌入式设备上使用常见的应用级网络协议，如何配置和生成与系统精确匹配的 Linux 内核；最后介绍 Linux 设备驱动的相关知识。

第三部分介绍 Linux 的组件和工具，包括 BusyBox、u-boot boot loader、git 等工具的安装、配置和初始化。

每部分内容都有相关的背景介绍，对于初级开发应用中可能遇到的问题也有详尽的说明，并且每章最后都有关于参考资料的介绍，对深入学习非常有帮助。

对于初级开发者来说，本书是一本难得的指导书。

由于时间仓促，不足之处欢迎批评指正。

欢迎邮件沟通：zhouyan05@hotmail.com。

周 艳

2014 年 11 月

前 言

二十年前的今天，佩铂军士在教乐队表演。

——约翰·列侬，佩铂军士的孤独之心俱乐部乐队

来自：torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds)

新闻组：comp.os.minix

主题：你最想在 minix 中看到什么？

摘要：关于我的新操作系统的一个小调查

消息 ID：<1991Aug25.205708.9541@klaava.Helsinki.Fi>

时间：格林尼治时间 1991 年 8 月 25 日 20:57:08

组织：赫尔辛基大学

使用 minix 的朋友们，大家好！

我正在为 386（486）AT 系列编写一个（免费的）操作系统（这仅仅是业余爱好，不会太大，也不会像 GNU 那样专业）。我从 4 月份开始酝酿，现在快准备好了。我希望得到任何关于 minix 的反馈，包括你们在使用过程中喜欢和不喜欢的地方。因为我的操作系统和它多少有点类似（比如文件系统具有相同的物理布局（出于实用的原因））。

我目前已经移植了 bash(1.08) 和 gcc(1.40)，而且进展好像很顺利。这意味着我将在未来几个月内把这个系统推向应用，所以我想知道大多数人希望系统具有什么特性。欢迎任何建议，但是我不能保证都能实现：-）

Linus(torvalds@kruuna.helsinki.fi)

补充一下，任何 minix 代码都是免费的，而且具有多线程的文件系统。

它不是便携的（如使用 386 任务切换等），很可能永远不能支持除了 AT 硬盘以外的东西，这就是我目前能达到的状态：-(

在当时刚出现的只能处理文字的互联网上，Linus Torvalds，一个芬兰的大学生，用这则信息开启了一场革命。当我在 2011 年年末写下这些文字时，Linux 已经庆祝了它的 20 周年纪念日。我怀疑，Linus 当时并没有想到他的“爱好”项目会很快变成一个功能强大的 32 位和后来的 64 位操作系统，甚至可以与微软有力地竞争。

毫无疑问，20 年来，Linux 走过了很长的一段路。1991 年 9 月 17 日首次发布的 0.01 版本包含 88 个文件，总共 278KB。最新发布的 Linux 版本包括近 40 000 个文件，超过 400MB。据说，全球几千个程序员都不同程度地参与到了 Linux 的开发中。

自 2006 年本书的第 2 版出版以来，Linux 世界又发生了很多变化，这也是本书再版的一个直接原因。Linux 内核持续改进，为了庆祝 20 周年，最近提高到了 3.0 版。2007 年年底，谷歌引入了安卓系统。据统计，截至 2011 年 8 月，安卓占有智能手机市场 48% 的份额，每天有 500 000 台设备被激活。

在本书第 1 版的前言中我就承认，我没有真正喜欢过 UNIX，因为它有意设计得晦涩而且很难使用。一开始，Linux 在这方面几乎没做任何改变，我也仍然坚持这种几分喜爱几分讨厌的态度。

但是，当 Linux 逐渐开始为桌面处理世界的黄金时代做好准备时，它在我工作的嵌入式领域和网络服务器方面都取得了显著的成就。2007 年，LinuxDevices.com 报道了 Linux 正用于近 50% 的嵌入式项目，到 2012 年将有超过 70% 的相关发布产品。甚至微软公司也承认全球有 60% 的网络服务器运行 Linux 操作系统，与此同时只有 40% 运行 Windows。

Linux 确实很复杂，除非你已经是一个 UNIX 高手，学习起来才会很快。相关信息就在网上，但是通常它们既不易找到也不易读懂。市场上有数百本已出版的 Linux 书籍，覆盖了从初学者指南到内核内部工作原理的各个方面。目前为止甚至有不少书讨论过嵌入式和实时应用领域的 Linux。

我决定学习 Linux，一部分是因为我看到了这个新兴的市场机会，另一部分是因为这种开源的开发模式激励了我。仅仅是出于对它的兴趣，全世界程序员的创意都对这个高度成熟的操作系统做出了贡献。拥有完整的源代码不仅允许你随自己意愿进行改动，也让你（至少在理论上）能理解代码是怎么工作的。

开源有潜力成为社会商业模式转换方面的一个重要范例，因为它表明在解决问题方面，合作可以和竞争同样有效。而且就在我怀疑是否有人能通过 Linux 得到惊人的财富时，许多公司已经证明针对 Linux 建立一个赚钱的商业模式是可能的。

读者和阅读准备

我写本书的目的是写一本在我初学 Linux 时渴望拥有的书。因此，基于这样的想法，本书是直接面向两种不同读者的：

1）主要读者是嵌入式程序员，他们需要得到在嵌入式领域的 Linux 指导。这也是我初学 Linux 时的状况和我开始学习 Linux 的原因，因此，这似乎是架构本书的一个合理方法。

2）另一类读者是 Linux 程序员，他们需要得到嵌入式和实时编程概念的指导。

因此，尽管本书采用了一个新的角度，但每类读者都很可能看到一些熟悉的内容。

本书不是初学者指南。我假设你已经成功安装了一个 Linux 系统而且已经使用过一些功能。你知道怎么登录，已经试验过一些命令实用程序，而且可能已运行了一个 GUI 桌面。尽管如此，第 2 章也会介绍安装的整个过程，第 3 章是嵌入式和实时程序员感兴趣的 Linux 的

一些特点和属性的简要介绍。

本书分为三部分。第一部分主要是介绍性的，为第二部分打基础；第二部分讨论在交叉开发环境中的编程应用；第三部分是前面介绍过的一些组件和工具的细节性描述，也是对Linux的实时性能的考量。

这里并没有说你不能通过读一本书去学习编程，你必须要这样做。这也是为什么本书设计为实用指导手册的原因。你将要在Linux工作站上安装许多软件包，你可以在本书英文版的网站上找到范例代码，www.elsevier.com/。

嵌入式编程通常意味着目标机器是独立的且与工作站开发机器不同。本书中，我们主要的目标环境是一个基于ARM的单板计算机，该硬件将在第5章中介绍。这是一款应用广泛且相对便宜的器件，它具有嵌入式实时产品需要的常规处理能力。但是即使你不购买电路板，我们也会介绍一种仿真方法，让你具有玩转工作站的一些基本概念。

个人建议

和大多数的计算机使用者一样，不管怎样，我多年来一直使用Windows，而且说实话，现在我仍然在日常的计算处理中使用Windows。但是同样，在那之前，我已经很熟练地使用DOS了，甚至在此之前，我已在使用RT-11、RSX-11和VMS。所以我并不是不懂命令行编程，事实上，在Windows 95时代以前，在我最终把WIN加入AUTOEXEC.BAT文件以前也已经用了好几年。

尽管如此，硬核UNIX程序员仍然认为GUI是为外行人设计的。他们很自豪地用命令行完成任何工作。但是不管你怎么做，我是喜欢GUI的。的确，命令行仍然有其地位，尤其是对shell脚本和makefile来说。但是对于在文件层次间切换和完成简单的文件操作，如移动、复制、删除、重命名等，简单的拖放操作完胜晦涩的UNIX shell命令行。我也拒绝使用基于文本的编辑器，如vi和emacs，不过我最近也开始接受vim。如果你能记住所有难懂的命令行，它们的确功能强大。但是无论如何，我喜欢所见即所得的编辑器。

我喜欢的GUI是KDE桌面环境。它包含所有必要的提示，包括很好的彩色语法编辑器，更不用说完整的办公包和个人高效工具包了。KDE是“K Desktop Environment”的缩写，它包含在最商业化的Linux发行版中。当然，你可以自由使用任何你觉得最习惯的开发环境来试验本书中的例程。但如果你是Linux新手，建议使用KDE，这也是本书所有示例的工作环境。

本书结构

本书由以下三个主要部分组成。

第一部分——第1～6章。这部分是入门介绍和使用指南。如果你还没有Linux，你将

需要安装一个。对于不熟悉 UNIX/Linux 系统的读者来说，这是对 Linux 本身的介绍。你需要设置和连接目标板，这在第二部分研究交叉平台应用开发的时候会用到。最后，你将安装并熟悉各种交叉开发工具，包括 Eclipse，它是一个开源集成开发环境。

第二部分——第 7 ～ 12 章。探讨交叉平台环境的应用编程。我们将着眼于从用户空间、调试技术和初步调试的高级仿真方面介绍硬件，介绍 Posix 线程、网络编程和安卓，以及内核级的编程和设备驱动。

第三部分——介绍嵌入式 Linux 程序员工具包中的各种组件和工具。我们将介绍其中的 u-boot boot loader、BusyBox、Buildroot 和 OpenEmbedded，最后将集中探讨 Linux 在实时性方面的优化处理。

好了，那就让我们开始吧。和我一起坐上充满了紧张和兴奋，有时很颠簸，但终将得到快乐和回报的过山车，进入激动人心的嵌入式 Linux 世界吧。

目 录

第一部分

入门指导

第 1 章

嵌入式和实时空间

如果你想周游世界并应邀去许多不同的地方演讲，那么写一个 UNIX 操作系统就可以了。

——Linus Torvalds

1.1 什么是嵌入式

在一个聚会上，当一个有吸引力的异性走近你，问你的工作时，你应该轻描淡写，说得越少越好，但最终谈话还是会落在你为嵌入式系统写软件这个事实上。在这位新相识开始扫视周围，找到一个律师或医生谈话前，你最好用最引人注意的解释说明到底什么是嵌入式系统。

我通常会说嵌入式系统就是一个内置计算机的设备，但是该设备的使用者不必知道或者特别在意那个计算机。它是隐蔽的。我通常以汽车里控制发动机的计算机为例来说明。你不觉得开车有什么不同，那是因为发动机是计算机控制的，而且汽车里还有很多计算机，有的控制防震刹车片，有的决定什么时候打开气囊，另外一些在交通拥堵的早晨给坐在车里的你提示信息和提供娱乐。事实上，今天普通的车都比 20 世纪 70 年代的登月飞行器有更强的处理能力。甚至你的手机也比登月飞行器处理能力强大。

你可以继续举出很多例子，嵌入式计算机的应用比个人电脑（PC）要多得多[⊖]。事实上，最近的市场数据显示，个人计算机使用的微处理器芯片只占每年市场份额的 2%。普通的房间即使没有个人电脑，也至少有几十个嵌入式计算机。

从编程的角度来说，嵌入式系统与传统的桌面应用有很多显著区别。比如，多数桌面应用处理的都是相对可预见的一组输入输出（I/O）设备——磁盘、图形显示、键盘、鼠标、声卡和网络接口。操作系统通常都可以很好地支持这些设备，应用程序员不需要特别关注这些。

⊖ 当前，多数人至少有模糊的嵌入式计算机的概念，因为他们至少都使用一个这样的计算机——手机。

但与之相反，嵌入式系统通常比典型的桌面计算机包括种类更多的 I/O 设备。一个典型的系统可能包括各种用户 I/O，如开关、按钮和增加了各种显示方法的触摸屏。它可能有一个或多个通信通道，或者是异步串口、通用串行总线（USB）或网络端口。它可能通过模数转换器（A/D）和数模转换器（D/A）完成数据采集和控制。这些设备几乎都没有应用程序员所熟悉的操作系统的支持。因此，嵌入式系统程序员通常都需要直接处理硬件。

嵌入式设备通常受资源的严格限制。尽管一个典型的 PC 目前已发展到有 4GB 的 RAM 和几百 GB 的磁盘，但嵌入式设备通常只有几个 MB 的 RAM 和非易失存储器。这些都对编程者的创造力有很高要求。

1.2　什么是实时

实时的概念更难解释。实时的基本含义是我们期望计算机对它的环境即时响应。但是什么是“即时”呢？有人认为实时意味着真的很快，这并不完全正确。实时仅仅意味着在系统运行的环境里足够快。如果我们谈论的是控制汽车发动机的计算机，那就是快！这个计算机需要决定每次发动机转一圈时对燃料流量、点火时间的控制。

另一方面，考虑一个或多个计算机控制的化工厂，计算机系统负责控制过程和检测潜在的破坏性故障。因为化学过程的时间常数最少在几秒到几分钟之间，所以可以假设计算机系统可以有足够的时间处理任何故障以避免造成损失。

但是假设当故障发生时，计算机正在打印一份关于上周生产情况的长篇报告或正在处理工资单，那么它对潜在的紧急情况的反应有多快呢？

实时处理的本质不仅在于计算机对它的环境有足够快速的响应，而且在于足够快的可靠响应。发动机的控制计算机必须在发动机转动的每一圈都调整燃料流量和点火时间。如果有所延迟，发动机就不能正常工作。化工厂的控制器必须有足够的时间正确检测和响应异常情况以避免事故，如果不能，这个控制器就是无效的。

我觉得下面这句话说得很恰当：

> 实时系统处理的正确性不仅要求处理的逻辑正确，也要求在规定时间内有结果。如果系统的时间约束不能被满足，那么可以说系统就失效了。
>
> ——Donald Gillies，实时处理常见问题

所以实时编程的艺术就是在随机的异步事件中，设计能可靠地满足时间约束的系统。的确，这说起来容易做起来难，目前有很多研究实时系统原理的文献和开发工作。

1.3　为什么 Linux 适用

Linux 模仿 UNIX 的基本架构，在其基础上发展成为一个通用的操作系统。没有人认为

UNIX 适用于嵌入式或实时操作系统（RTOS）。它太大了，占用太多的资源，而且其调度原则是基于顺序而不是优先级。因此，简而言之，它与嵌入式操作系统的要求完全是背道而驰的。

但是 Linux 有 UNIX 系统的早期版本所欠缺的几样东西。它是免费的，而且你可以得到源代码，并且有一个拥有大量热情的 Linux 开发者和使用者的大社区。你在使用中所面临的问题总有人正遇到或曾经遇到过，所有这些都在网络上，窍门就是找到它们。

1.3.1　开源

Linux 是在自由软件基金会（Free Software Foundation，FSF）倡导的开源软件的宗旨下发展起来的。开源的理念非常简单，即软件应该免费获取以用于使用、修改和复制。这个理念在建立了以太网和万维网的技术文化下发展了 20 多年，近年来开始向商业领域发展。

有许多关于开源软件的误解。也许对它是什么的最好解释就是从它不是什么开始。

- **开源不是共享**。使用共享软件的前提是付给版权持有者一定的费用。共享软件通常以一定的免费形式发布，或者时间受限或者功能受限。要得到完整版，就必须付钱。但与之相反，开源代码可以免费获取，你没有任何为之付钱的义务。
- **开源不是公共**。根据定义，公共代码是没有版权的。开源代码的作者拥有版权，他在开源软件协议的框架下发布软件。版权持有者授权你可以在协议的框架内使用代码，如果你不遵守协议，版权持有者有权要求你停止使用代码。
- **开源不一定免费**。没有任何强制规定要求付款给开源软件代码，但不排除收取打包和发布费用。许多公司都在经营特定的打包 Linux 分发产品。

为什么要为免费得到的东西付钱？想必是因为任何东西只要存在，你就可以从供应商那里得到一些支持。当然，支持的质量很大程度上取决于供应商。

所以免费意味着可以自由使用代码但不意味着零花费。想想“免费演讲”和“免费啤酒”的区别你就明白了。

开源代码是：

- **服从开源协议框架**。在许多案例中都是 GNU 通用公共协议（GPL）[⊖]。
- **经过严格的同行评议**。作为一个开源程序员，你的代码就放在那里，任何人都可以看到。开源社区非常严格，开源代码会经历广泛的测试和同行评议。这是一个适者生存的竞争过程，最后只有最好的代码能够存在。“最好”是一个主观的术语。它可能是最好的技术解决方案但同时也可能可读性很差。
- **高度的颠覆性**。开源运动颠覆了主流模式，主流模式认为软件之类的知识产权应该受到保护，所以你可以通过这个赚钱。与之相反，开源的理念是软件应该免费提供给每个人，这样可以最大化地利于社会的发展。如果你可以通过这个赚钱，那挺好的，但

⊖ GNU 是一个自参照缩写，意思是“GNU 非 UNIX”。GNU 的项目由 Richard Stallman 在 1983 年发起，旨在打造一个“完整的兼容 UNIX 的系统”，完全由免费的软件组成。

这不是初衷。FSF 的创始人 Richard Stallman 一直大力倡导软件不应该属于某个人（见附录 C）。

因此当微软公司视开源为他们商业模式的一个严重威胁时，这就毫不奇怪了。微软的发言人一直将开源归为“非美式的”。而另一方面，许多开源软件的领头供应商一直让他们的程序员和工程师为开源社区添砖加瓦。这不仅仅是公益，更是一个很好的商业模式！

1.3.2　移植和定制

Linux 最初是为 Intel 的 x86 系列处理器开发的，而且多数正在进行的内核开发工作也还是针对 x86 系列的。尽管如此，Linux 内核的设计将有所不同，以前的代码是依赖于处理器的，且对于每个不同架构的处理器都需要改动，但开源代码则只要进行简单的重新编译就可以导入一个新处理器。因此，Linux 可以导入多数 32 位甚至 64 位的处理器，处理器架构包括：

- Motorola 68k 和它的很多变体
- Alpha
- PowerPC
- 高级 RISC 机器（ARM）
- Sparc
- MIPS

这只是很多流行的处理器中的一些。所以，无论嵌入式项目中要用什么样的 32 位架构，总有 Linux 可用，也总有很多的开发者在支持它。

一个典型的桌面 Linux 的安装运行需要 10 ～ 20GB 的磁盘空间，并需要 1GB 的 RAM 来较好地运行。但是嵌入式的目标板通常只有 64MB 或更少的 RAM 和也许是 128MB 的 flash ROM 存储器。幸运的是，Linux 是高度模块化的。10GB 的内容大都包括文档、桌面实用程序、游戏等选项，这些在嵌入式目标板中不是必要的。如果资源受限的话，生成一个功能完整的只占用不超过 2MB flash 存储器的 Linux 系统是不难的。

内核本身是高度可配置的，包括一些合理的用户工具，允许在应用中去掉某些不需要的内核功能。

1.4　哪里用嵌入式 Linux

任何地方都在用。2005 年 7 月，LinuxDevices.com 网站列出了超过 300 个运行 Linux 的商业产品。它们包括手机、个人数字助手（PDA）及通过路由器和网关使用的其他手持设备；精简客户端、多媒体设备、电视机顶盒、机器人和加固的 VMEbus [⊖]；适用于军事和控制应用的机箱。这些只是 LinuxDevices 网站的编辑们正好知道的产品。

⊖ 即 VERSA Module Eurocard bus，是工业计算机打包和互联标准。