



嵌入式 Linux

—— 硬件、软件与接口

Embedded Linux

Hardware, Software, and Interfacing

[美] Craig Hollabaugh 著

陈 雷 钟书毅 等译

魏永明 审校



电子工业出版社
Publishing House of Electronics Industry
<http://www.phei.com.cn>

嵌入式 Linux

——硬件、软件与接口

Embedded Linux
Hardware, Software, and Interfacing

[美] Craig Hollabaugh 著

陈 雷 钟书毅 等译
魏永明 审校

电子工业出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书通过一个冬季旅游胜地自动化管理项目实例,从软件、硬件和接口的观点介绍嵌入式Linux。引入项目需求后,作者讲述了开发环境的建立,接着用一系列软硬件接口实例展示了如何使用异步串行通信、PC并口、USB、内存I/O、同步串行通信以及中断,等等,最后介绍了将前面所有的工作有机地组织在一起的系统集成过程。本书以实际应用为导向,书中整个项目的实施过程和软硬件接口实例都极具实践指导意义。

本书实用易读,适合硬件工程师、软件开发人员、系统集成人员或产品经理以及其他研究嵌入式Linux应用的人士阅读。

Authorized translation from the English language edition, entitled *Embedded Linux: Hardware, Software, and Interfacing*, ISBN: 0672322269 by Craig Hollabaugh, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2002.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright © 2003.

This edition is authorized for sale only in the People's Republic of China excluding Hong Kong, Macau and Taiwan.

本书中文简体专有翻译出版版权由Pearson教育集团所属的Addison-Wesley授予电子工业出版社。其原文版权及中文翻译出版版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

此版本仅限在中华人民共和国境内(不包括香港、澳门特别行政区以及台湾地区)发行与销售。

版权贸易合同登记号:图字:01-2002-5191

图书在版编目(CIP)数据

嵌入式Linux——硬件、软件与接口/(美)霍拉鲍夫(Hollabaugh, C.)著;陈雷等译.

-北京:电子工业出版社,2003.11

书名原文:Embedded Linux: Hardware, Software, and Interfacing

ISBN 7-5053-9297-2

I. 嵌... II. ①霍... ②陈... III. Linux 操作系统 IV. TP316.89

中国版本图书馆CIP数据核字(2003)第100114号

责任编辑:赵红燕 贺瑞君

印刷者:北京兴华印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:19.25 字数:479千字

版 次:2003年11月第1版 2003年11月第1次印刷

定 价:35.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。

联系电话:(010)68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

嵌入式系统的时代已经来临,尽管它一度沉寂。现在,我们的日常生活中到处都有嵌入式系统的身影:带有GPS系统的汽车、智能化的手机、数码相机、DVD机、PDA等等,嵌入式系统的工业应用则更为广泛。在这个万人瞩目的朝阳产业中,后起之秀Linux因为其源代码的开放性和强大的功能,日益显出不可替代的优势。

尽管业界的实际应用已经很深入,但嵌入式Linux方面的信息资料却远没有Linux其他方面的全面、集中,而国内这方面的开放性资源更为稀少。Craig Hollabaugh博士撰写的“Embedded Linux: Hardware, Software, and Interfacing”正是大家所需要的,而我们也有幸能够及时为广大的中文读者翻译这本难得的实用读本。

本书没有停留在讲述原理的层面,而是以一个利用嵌入式Linux开发的实际项目为背景,通过讲述该项目中涉及到的各个嵌入式Linux技术主题,较全面地向读者展现了利用嵌入式Linux开发项目的过程。本书讲解由浅入深、对问题的分析十分透彻,实践性很强,并且对一些常见问题的解决方案以及一些技巧进行了归纳,提供了嵌入式Linux开发过程中涉及的常见技术的网络资源。书中给出了很多简单而又易于扩展的源代码,其中包括设备驱动程序等C程序以及bash脚本。读者只需具备基本的数字逻辑电路、C语言以及Linux系统管理知识即可。通过阅读本书,读者可以一步步地真切了解到嵌入式Linux开发的具体过程。

本书主要由陈雷、钟书毅翻译,魏永明审校,另外参与本书翻译的还有姜君、张云帆、张晓明、窦露、康慨和耿岳。鉴于时间仓促以及译者水平有限,不足之处在所难免,欢迎广大读者批评指正。

序 言

嵌入式系统在信息技术方面曾经一度处于停滞状态,而现在已经成长为一个数十亿美元的产业,正日益渗透和改变着我们的日常生活。在世界范围内,“智能设备”逐渐进入每个人的日常活动之中:驾驶电子控制的汽车或乘坐数字化的市郊火车上班,使用移动电话,在办公室使用复印机等设备,出席数字电话会议,然后回家观看有线电视或DVD设备播放的影片。环顾四周,我们将发现周围总有嵌入式计算机影响着我们每一天的生活。

Linux正在这个嵌入式和普及计算的世界中前进。嵌入式系统曾被专用和难懂的操作系统、工具 and 应用程序所占据,现在,嵌入式设计却大大受益于开源(open-source)软件(特别是Linux)的透明性和强大的功能。甚至保守的估计都表明,2002年末,嵌入式Linux将获得新的嵌入式设计一半以上的份额。

作为MontaVista Software公司的战略市场/嵌入式Linux开拓部门主管,我很高兴有机会和成千上万的嵌入式开发者交流,了解他们研究的各种项目。在此情况下,我在2001年纽约的LinuxWorld大会上首次遇见了Craig Hollabaugh。在他撰写本书期间,我曾有幸多次和他见面并交谈。

我发现Craig的方法很可行也很全面。他选择的应用于Silverjack(虚构)冬季度假胜地的应用程序,既是嵌入式应用空间(控制器和仪器)的最基本部分,也涉及当今普及计算(嵌入式Web接口、多媒体和信息传送)中最热门的新领域。Craig关于可用的嵌入式硬件和关键接口技术的全面评述,以及对嵌入式Linux开发过程按部就班的阐述,为热衷于此的系统设计者提供了宝贵的指南。

Craig的“开拓者”工程师团队面临着获取和使用嵌入式Linux开发工具,以及部署各组成部分的艰巨任务。Silverjack方案和工程工作的细节,反映了许多开发者在他们首次面对嵌入式Linux工程时的经历。任务很艰巨,但回报是巨大的。诸如《嵌入式Linux》这样的书籍,还有嵌入式Linux联盟、EMBLIX以及LinuxDevices这样的组织,加上嵌入式Linux平台厂商的努力,将为成功地开发和部署下一代智能的、基于Linux的设备铺平道路。

William Weinberg

MontaVista 软件公司 战略市场/嵌入式Linux开拓部门主管

前 言

这是一个令人激动的时代。互联网技术的高速增长已经改变了信息这一概念。人们每天24小时依靠网络设备进行商务和个人的活动。随着时间的推移,这种依赖的程度还将进一步增长。当今的网络设备,例如服务器、路由器和桌面计算机,构成了互联网的基础体系结构。未来的设备将能控制房间的温度,维持商业库存水平并监控汽车交通状况。每年生产的处理器中只有几十亿进入了桌面或服务器计算机,余下的处理器监视和控制着世界的其他方面。

个人和商业都需要智能和联网的设备来改善人们的生活和效率。这些设备必须易于操作、可靠而且价格低廉。这正是Linux流行的原因。技术的进步加上有竞争力的价格,使Linux能够从桌面和服务器环境转移到嵌入式设备中。Linux较为可靠并提供丰富的功能,而且是开源代码,具有公认的良好记录,这些特色使之非常适合于嵌入式系统。本书将介绍创建一个嵌入式Linux基础环境的相关信息,然后读者就可以在自己的设计中扩展该基础环境。

本书的好处

当我开始研究本书的相关资料时,我以为在线的嵌入式Linux信息与其他的Linux信息一样——非常丰富,而且文档也应该很全面,但我很快发现,这样想是错误的。实际上,嵌入式Linux的文档稀少、零散、不完整而且有些过时。这是令人沮丧的,但同时对我也是一种激励。因为一方面可能缺少足够的信息,但另一方面也说明了本书的必要性。人们正在用Linux设计嵌入式产品,信息和知识也已存在,只是没有集中起来而已。

作为一名教师,我确信在有相应的实例时,学生能够最好地理解和记住所学理论的概念和思想。当学生看到一个概念付诸实施时,无论是一次机械臂的移动,一次电压表的读取,还是一个示波器波形,他们很可能就在稍后用这一概念解决自己的问题。为此,本书引用大量逐步的实例。借助它们,读者能够将所学知识应用到自己的设计中。

本书内容

本书包括一系列完整的真实世界的接口实例,它们用来从硬件和软件的观点介绍嵌入式Linux。创建一个嵌入式Linux开发环境之后,将逐步介绍每一个硬件和软件的接口实例,包括使用异步串行通信、PC并行端口、USB、内存I/O、同步串行通信以及中断等等。稍后将使用系统集成将所有的接口实例整合在一起。这些素材都是通过一个称为开拓者项目(Project Trailblazer)的冬季度假胜地自动化工程展示的。可以在www.embeddedlinuxinterfacing.com上找到本书的源代码和脚本。

下面简要概述每一章。

第1章阐述了Linux作为一个嵌入式操作系统的历史,以及在产品设计中使用开源软件的意义。

第2章介绍一个称为开拓者项目的冬季度假胜地自动化工程,规划出一系列的项目需求。

该项目及其需求构成了本书中接口实例的基础。

第3章介绍平台选择的过程。工程师选中了4个目标机，使用x86、StrongARM和PowerPC处理器。随后讲解如何创建名为tbdev1的嵌入式Linux开发工作站。全部开发工具都要安装或编译，包括用于StrongARM和PowerPC处理器的交叉编译工具链。

第4章介绍从加电到出现bash提示符的Linux引导过程。每一个目标机均以Linux 2.4版本引导，使用最小的根文件系统。

第5章介绍如何配置gdb和gdbserver，从而通过以太网对目标机进行调试。一个交叉编译版本的helloworld将远程执行和调试。

第6章讲述用于控制端口信号和缓存的Linux串行端口设备驱动。一个RFID标签阅读器、一个LCD显示器及其控制电路连接到Linux串行端口。

第7章讲述AC电路到x86目标机并行打印端口的连接。此外，还介绍了使用/proc目录项、名为helloworld_proc_module的自定义设备驱动。

第8章讲述用于可视输入及音频输出的摄像头和扬声器到目标机的USB端口的连接。

第9章讲述AC电路到StrongARM和PowerPC目标机的CPU总线的连接。

第10章讲述SPI和I2C连接和通信。连接一个低成本的SPI温度传感器和I2C LED显示驱动器到目标机。

第11章讲述Linux的时间调配机制以及每个目标机的平均中断响应时间的测量。开发出一种精确到1 ms的事件计时器，用于测量赛事时间。

第12章讲述开拓者项目数据库的创建。首先开发出目标系统和服务器的bash脚本，使用该数据库进行温度、图像和认证数据的收集和分发。

第13章总结接口工程并讨论本书其他章节未涉及的嵌入式Linux主题。

读者对象

硬件工程师、软件开发人员、系统集成人员或产品经理，以及正在研究嵌入式Linux接口应用的人员，都适合阅读本书。书中所有的接口实例都很简单，只要对数字逻辑设计、C和bash编程以及Linux系统管理有基本常识即可理解。

目 录

第 1 章 嵌入式 Linux 简介	2
1.1 为什么 Linux 广受欢迎	2
1.2 什么是嵌入式系统	3
1.3 什么是实时	3
1.4 开放源代码的含意	4
1.5 小结	5
补充读物	5
第 2 章 系统体系结构	6
2.1 开拓者项目简介	6
2.2 Silverjack 度假胜地的布局	7
2.3 开拓者项目的需求	8
2.3.1 安全需求	8
2.3.2 运营需求	9
2.3.3 供应商需求	10
2.4 开拓者项目的系统体系结构	10
2.5 小结	11
第 3 章 选择平台和安装工具集	12
3.1 信息来源	12
3.2 开拓者项目的战略方向	13
3.3 构建 tbdev1——嵌入式 Linux 开发工作站	14
3.3.1 开始	15
3.3.2 安装 Linux 操作系统	16
3.3.3 安装本机 GNU 工具链和其他应用程序	20
3.3.4 创建用于交叉编译的 GNU 工具链	20
3.4 小结	44
补充读物	44
第 4 章 引导 Linux	45
4.1 目标机的平台引导需求	45
4.2 Linux 的引导过程	46
4.3 Linux 的根文件系统	47
4.3.1 init 所需的文件	48
4.3.2 bash 所需的文件	49
4.3.3 根文件系统的二进制文件：编译还是下载	50
4.4 创建根文件系统	61

4.4.1	决定要使用的包	61
4.4.2	创建根文件系统的过程	62
4.4.3	运行 buildrootfilesystem	63
4.5	安装 TFTP 服务器	66
4.6	安装 minicom	67
4.7	引导 Embedded Planet RPX-CLLF	67
4.7.1	Embedded Planet RPX-CLLF 的目标 PBR 回顾	71
4.8	引导 Brightstar Engineering MediaEngine	73
4.8.1	Brightstar Engineering MediaEngine 的目标 PBR 回顾	76
4.9	引导 Tri-M MZ104 和带闪存 IDE 驱动器的 COTS PC	78
4.9.1	闪存 IDE 技术	78
4.9.2	准备 Tri-M MZ104 和 COTS PC	79
4.9.3	引导 Tri-M MZ104 目标系统	82
4.9.4	Tri-M MZ104 的目标 PBR 回顾	83
4.10	不同引导方式之间的比较	85
4.11	小结	86
	补充读物	86
第 5 章	调试	87
5.1	gdb 简介	87
5.2	本机调试	88
5.3	远程调试	91
5.4	通过网络挂装根文件系统	94
5.4.1	配置 NFS 服务器	95
5.4.2	配置目标内核	96
5.5	小结	99
	补充读物	99
第 6 章	异步串行通信接口	102
6.1	开拓者项目的异步串行硬件开发环境	102
6.1.1	目标 EIA/TIA-232-E 兼容性	103
6.2	Linux 串行通信	103
6.2.1	使用 setSerialSignal 设置串口控制信号	104
6.2.2	使用 getSerialSignal 读入串口控制信号	107
6.2.3	文件 open 系统调用如何影响 DTR 和 RTS 信号	109
6.2.4	使用 querySerial 为 bash 脚本提供串行通信功能	110
6.3	小结	114
6.4	补充读物	115
第 7 章	并行接口	116
7.1	使用并口进行控制	117
7.2	使用端口 I/O 控制标准并口	120
7.2.1	使用端口 I/O 监测缆车运行状况	120

7.2.2	使用端口 I/O 控制造雪设备	123
7.3	使用 ppdev 控制标准并口	125
7.4	开发定制的设备驱动程序	128
7.4.1	理解 helloworld_proc_module	129
7.4.2	在 MediaEngine 上编译、插入模块并测试 helloworld_proc_module	134
7.5	使用定制设备驱动程序 liftmon_snowcon 控制标准并口	135
7.5.1	增强 helloworld_proc_module 以创建 liftmon_snowcon	136
7.5.2	在 MZ104 上编译、插入和测试 liftmon_snowcon	146
7.6	小结	149
7.7	补充读物	149
第 8 章	USB 接口	150
8.1	什么是 USB	150
8.2	开拓者项目的 USB 硬件	152
8.2.1	USB 音频: Yamaha YST-MS35D USB 扬声器	152
8.2.2	USB 视频捕捉: Kensington VideoCAM super-VGA PC 摄像头	156
8.2.3	USB 存储: SanDisk USB SmartMedia 读卡器	160
8.3	小结	164
8.4	补充读物	164
第 9 章	内存 I/O 接口	165
9.1	硬件设计过程	165
9.2	为 MediaEngine 开发缆车监测和造雪设备控制功能	166
9.2.1	设计 MediaEngine 的外部接口硬件	166
9.2.2	MediaEngine 的内存映射空间寻址	168
9.2.3	MediaEngine 的寄存器基址寻址	169
9.2.4	为 MediaEngine 配置内存控制器	169
9.2.5	为 MediaEngine 分配输出模块启用信号	169
9.2.6	为 MediaEngine 配置 I/O 端口控制器	169
9.2.7	为 MediaEngine 编写 helloworldbit 测试驱动程序	170
9.2.8	为 MediaEngine 编写 liftmon_snowcon 设备驱动程序	174
9.3	为 RPX-CLLF 开发缆车监测和造雪设备控制功能	184
9.3.1	为 RPX-CLLF 设计外部硬件接口	185
9.3.2	为 RPX-CLLF 寻找内存映射空间	185
9.3.3	为 RPX-CLLF 寻找寄存器基址	187
9.3.4	为 RPX-CLLF 配置内存控制器	187
9.3.5	为 RPX-CLLF 分配输出模块启用信号	188
9.3.6	为 RPX-CLLF 配置 I/O 端口控制器	188
9.3.7	为 RPX-CLLF 编写 helloworldbit 测试设备驱动程序	188
9.3.8	为 RPX-CLLF 编写 liftmon_snowcon 设备驱动程序	192
9.4	小结	202
9.5	补充读物	202

第 10 章 同步串行通信接口	203
10.1 温度测量和显示	203
10.2 SPI 通信与 LM70	204
10.2.1 将 LM70 连接到 x86 的并行打印口	205
10.2.2 将 LM70 连接到 MediaEngine	212
10.3 与飞利浦 SAA1064 进行 I2C 通信	219
10.3.1 将 SAA1064 连接到 x86 并行打印端口上	219
10.3.2 将 SAA1064 连接到 RPX-CLLF	224
10.4 小结	234
10.5 补充读物	235
第 11 章 使用中断计时	236
11.1 Linux 计时源	236
11.2 测量中断响应时间	237
11.2.1 在 MZ104 上测量中断响应时间	238
11.2.2 在 MediaEngine 目标机上测量中断响应时间	243
11.2.3 在 RPX-CLLF 目标机上测量中断响应时间	250
11.2.4 对中断响应时间测量的总结	255
11.3 比赛计时器的实现	256
11.3.1 使用任务队列处理比赛计时器的中断	257
11.3.2 使用系统定时器进行比赛计时状态的显示	258
11.4 小结	269
补充读物	269
第 12 章 系统集成	270
12.1 系统集成概述	270
12.2 安装系统集成应用程序	271
12.3 创建并测试开拓者项目的数据库	275
12.4 开发目标机和 CGI 集成脚本	278
12.4.1 采集和发布温度数据	278
12.4.2 采集并发布图像数据	284
12.4.3 采集游客标识信息并允许乘坐	289
12.5 小结	292
12.6 补充读物	293
第 13 章 最后的思考	294
13.1 GUI 开发	294
13.2 实时能力	294
13.3 嵌入式厂商简介	295
13.4 开拓者项目的硬件	296
13.5 小结	297
13.6 补充读物	297

第一部分

入门进阶

- 第 1 章 嵌入式 Linux 简介
- 第 2 章 系统体系结构
- 第 3 章 选择平台和安装工具集
- 第 4 章 引导 Linux
- 第 5 章 调试

第1章 嵌入式Linux简介

本章要点:

- 为什么Linux广受欢迎
- 什么是嵌入式系统
- 什么是实时
- 开放源代码的含意
- 补充读物

你是不是已经打算在自己的嵌入式系统中使用Linux了？如果是，很好！你不是惟一做此选择的人，全世界有成千上万的开发者和你一样。他们希望利用丰富的代码资源使自己的产品变得可靠、强壮而富有特色，同时还可以迅速投入市场。嵌入式Linux（即在嵌入式设备中使用Linux作为操作系统）现在已经非常普遍。1999年时，还很少有开发者在嵌入式设备中考虑使用Linux。然而在今天，Linux随时都有可能成为嵌入式操作系统市场的主导者。

两三年内，每年生产的数十亿个处理器中，将有相当大的一部分运行Linux。因特网对Linux大量的采用、技术的进步以及硬件（CPU）和存储器（闪存及RAM）价格的下降，将最终使嵌入式Linux成为一个切实可行的选择。1999年以来，嵌入式Linux的采用已经从零增长到占市场份额的第二位，仅次于Wind River¹。专家估计嵌入式Linux将于2002年底占据市场份额的第一位²。如果读者了解Linux实际上起源于许多年前一个学生所做的项目，将会更惊讶于其发展。随着越来越多的产品设计采用嵌入式Linux，Linux将成为最终的执牛耳者。

1.1 为什么Linux广受欢迎

微处理器工业的进步是使用Linux作为嵌入式操作系统的基础。在嵌入式设计中使用4位或8位微处理器的时代已经过去了。今天，32位微处理器和存储量以兆计的存储器不但使用简单，而且其价格也很有竞争力。硬件的不断革新向实时操作系统（RTOS，real-time operating system）厂商提出了巨大的挑战。RTOS厂商在有限的资源条件下做出的商业决策常常导致许多开发者得不到有效的支持。开发者总是希望在他们的嵌入式设计中使用最新、最好的硬件和软件，同时还要求高度的可靠性。桌面操作系统提供最新的软件，但是可靠性不强，而且需要更多的存储器和CPU资源。RTOS厂商提供的产品可靠性强，但是在不断变革的技术世界中却步履缓慢。这种情况下，Linux正可以发挥所长。

Linux具有很高的可靠性和效率，它在这方面的表现是公认的。其可公开获得的开源代码已经在性能、功能、驱动开发和可移植等方面历经了无数的改进、增强以及扩充。嵌入式系统的设计者喜欢开源软件，因为他们能够针对自己的设计修改子系统的代码。Linux支持可移植操作接口（POSIX，Portable Operations System Interface），该接口使开发者可以很容易地移植

现有的代码。Linux已经能成功地用在几种微处理器上,包括x86、SPARC、ARM、PowerPC、MIPS和SuperH。

最初,由于Linux的分布式开发途径及其缺乏支持结构的表象,开发人员很少考虑将其用于嵌入式操作系统。然而在1999年,Linux的支持结构发生了戏剧性的转变,这一点极大地归功于嵌入式Linux联盟(ELC, Embedded Linux Consortium)。ELC于2000年5月成立时仅有50名成员,发展到今天,其会员团体已超过125个。ELC的主要工作是标准化和改进Linux,使其适合于嵌入式应用。此外,还有一些独立的嵌入式Linux团体,他们不仅仅重新打包分发产品并销售CD,而且在非常广的范围内提供Linux产品以及设计和支持服务。开发人员使用嵌入式Linux可以设计可靠性很高的产品,并能应用最新和功能最强大的软件。

1.2 什么是嵌入式系统

如果分别让10位工程师给出嵌入式系统的定义,将得到10个不同的答案。一般来说,大部分的嵌入式系统执行特定的任务。我们假定最简单的嵌入式系统包括输入/输出功能,以及一些控制逻辑,该系统基于它的配置执行某些类型的功能。按照这个标准,可以认为一个包含实现控制逻辑的74123计数器以及一个状态机是一个嵌入式系统。也许可以补充说,该系统必须可通过存储在固件中的软件进行编程。这个新的嵌入式系统定义包括输入/输出(I/O),以及存储在系统固件中的控制逻辑。一个带有鼠标、键盘、网络连接并运行图形用户界面(GUI, graphical user interface)多任务操作系统的桌面计算机显然满足这些要求,但我们能认为它是一个嵌入式系统吗?

如果桌面计算机不是一个嵌入式系统,那么手持设备呢?它们有I/O功能,可以运行存储在固件中的控制逻辑。有人说,桌面计算机和手持设备都是通用计算设备,可以运行软件来执行许多不同的任务,与之不同的是,嵌入式系统(例如,洗碗机控制器或飞行导航系统)主要是为特定任务而设计的。这种特定的功能限定使嵌入式设备有功能上的惟一性。如果是这样,为什么一些嵌入式系统设计成具有附加的功能,如存储在非易失性存储器中的程序,并且具有运行可以完成原始设计范围之外的任务的多任务操作系统的能力呢?

在过去,区分嵌入式系统和通用计算机比现在简单得多。例如,可以很容易地区分出一个基于8051的T1分幅卡嵌入式系统和一台Sun UNIX工作站。而现在,从功能方面很难区分一台Sun工作站和一个包含PowerPC以及32 MB内存和16 MB闪存的机顶盒。这样的机顶盒可以运行带GUI的多任务操作系统,可现场升级,可以同时运行多个程序(如视频控制器、数字录像和Java虚拟机),还可以进行安全的因特网在线交易。很难判断这种机顶盒是否是一个嵌入式系统。显然,硬件性能的提升和价格的下降使通用计算机和嵌入式系统之间的界限变得很模糊,技术的进步使得我们很难定义什么是嵌入式。

1.3 什么是实时

如果问不同的工程师实时的概念,也将得到许多不同的答案,其中会充斥着诸如保证、硬、软、确定性、抢占、优先级转换、延迟、中断和调度等术语。linuxdevices.com的Rick Lehrbaum向7位专家提出了这个问题³。这些专家在某种程度上同意以下观点:一个硬实时系统保证事件响应的确定性,认为延迟的响应是系统错误;他们还在某种程度上认为,软实时系统试图使事

件的响应时间减到最小，而不将延迟的响应看做系统错误。专家经常在考查实时实现和系统性能时出现意见分歧。

Linux 的一个很明显的性能问题是常规的 Linux 允许内核和设备驱动禁用中断⁴，这样可能会严重地影响系统的响应速度。幸运的是，易得的开源代码已经提供了该问题的解决方案。这些解决方案遵循两种方法：一是通过增强内核使之可抢占，进而改进 Linux 的调度程序；另外一种则在一个小的 RTOS 内将 Linux 作为一个线程运行。毫无疑问，这两种方法各有优缺点，同时也有各自的支持者和反对者。

1.4 开放源代码的含意

开源软件，通过其易得性、可靠性以及支持方式，已经明显地威胁到了提供专有解决方案的商业软件厂商。将 Linux 用于嵌入式开发，威胁到了诸如 Microsoft 和 Wind River 这类公司未来的市场份额。在嵌入式行业内散布不安、不确定性和怀疑言论符合他们的最大商业利益^{5,6}。许多年来，无论是在桌面、服务器还是在嵌入式领域，微软一直不认为 Linux 是一种威胁。但这种情况已经改变，微软声明 Linux 现在是它的头号目标⁷。微软对 Linux 的关注，进一步提升了 Linux 作为嵌入式开发的一个可行选择的地位。许多开发者将开放源代码看做构建强有力产品的构件。微软则将 Linux 视为在嵌入式操作系统上对其未来统治地位的最大威胁。

开源软件的使用有特定的含意。在嵌入式设计中使用 Linux 时，要牵涉到开源软件和一个称为通用公共许可（GPL，General Public License）⁸的独特许可模式（这里不包括完整的 GPL 条款，但我们会给出影响读者及其嵌入式 Linux 设计的一些 GPL 要求）。

Linus Torvalds 最初是在 GPL 条款下发布 Linux 内核源代码的。这意味着可以自由使用和重新发布 Linux，而且不用缴纳版权费或许可费，但是必须向客户提供 Linux 的源代码。不允许销售 Linux，但可以销售发布媒介或任何做出的改进。例如，当购买一个 Red Hat 的 Linux 产品时，从 Red Hat 购买的并不是 Linux，而是 Red Hat 对 Linux 的增强。Red Hat 的增强包括易用的安装程序及其他一些吸引人的东西。GPL 声明所有的“衍生工作”也必须在 GPL 条款下发布。这意味着如果修改了 Linux 内核源代码的某些部分，这些修改同样要遵循 GPL。如果发行一个基于所修改内核代码的产品，不但要提供内核源代码，还必须以源代码的形式公布对其做出的修改。对于这一点，也许会有疑问：“如果从来不触及内核源代码而只是在产品中使用它，那么 GPL 又有什么影响呢？用户真的必须以源代码的形式提供其的产品吗？”答案取决于产品链接了什么（函数库）及其链接方式。

例如，假定你的嵌入式产品中包含一个新开发的设备驱动或内核模块，那么可以选择在编译时将设备驱动包括在内核的二进制代码中，也可以将该驱动作为可装载模块使用。如果创建和发布一个包含自己的设备驱动的内核，那么设备驱动代码自动包括在 GPL 条款适用的范围内，这时必须向客户提供该设备驱动程序的源代码。但是，在内核引导过程之后动态装载的设备驱动或模块却不适用于 GPL 条款，那么在这种情况下，不必公布自己的源代码。

注意：使用 Linux 和其他开源软件设计产品并不意味着必须使你的软件成为开源软件。可装载模块和 LGPL 库的使用可以保护个人的知识产权。

如果一个嵌入式产品只在用户空间执行，也就是说它没有内核代码和设备驱动，这种情况下，它不受 Linux 内核源代码 GPL 条款的影响。但是，如果在用户空间运行的应用程序链接的

库对应于GPL或另一个称为LGPL的许可模式,那么,这样的嵌入式产品可能会受到影响。LGPL最初称为库通用公共许可(Library General Public License),现在代表较为宽松的通用公共许可(Lesser General Public License)。该许可允许开发者链接函数库并使用其中的代码,但不必使这些代码自动包含在GPL条款之下。GNU在LGPL条款下发布了glibc⁹。如果嵌入式应用程序动态地链接glibc,则该应用程序不包括在LGPL条款之下,也就是说,不必公布自己的源代码。

开源软件的使用促进了创新,允许开发者快速设计出可靠和健壮的嵌入式应用程序。嵌入式设计基于开源软件,这并不意味着必须以源代码的形式发布自己的知识产权成果。可以开发出在内核引导过程之后装载的设备驱动或内核模块,以及链接到LGPL库的用户空间应用程序代码。

1.5 小结

Linux诞生于十多年前。无数的程序员对它进行了增强,赋予其更多的功能和特色,使它更加高效、可靠和强壮。今天,成千上万的商业活动依赖于Linux,进行日常的客户端和服务器的操作。Linux有公认的优良历程记录。微处理器和存储设备价格的下降,将使未来的嵌入式设计更加普遍地采用32位体系结构,而Linux已经移植到了这些体系结构上。目前,嵌入式Linux产品已投入市场。采用Linux的嵌入式系统设计者可以利用开源代码的易得性、POSIX兼容性、坚实的可靠性以及大量的应用程序基础,而且没有许可费用。这些好处,再加上价格优势,使得Linux成为嵌入式操作系统的一个出色解决方案。

补充读物

1. "Developer interest in Embedded Linux skyrockets," www.linuxdevices.com/news/NS2083407450.html, 2001.
2. Fernando Ribeiro Corrêa, "Intelligent devices: A new arena for Linux," www.olinux.com.br/artigos/279/1.html, 2001.
3. Rick Lehrbaum, "Real-time Linux—What is it, why do you want it, how do you do it?" www.linuxdevices.com/articles/AT9837719278.html, 2000.
4. Kevin Dankwardt, "Fundamentals of real-time Linux software design," www.linuxdevices.com/articles/AT5709748392.html, 2000.
5. Mike Downing, "Revision 3.0 of open-source GPL stirs concern in embedded space," http://icd.pennnet.com/Articles/Article_Display.cfm?Section=Articles&Subsection=Display&ARTICLE_ID=92350&KEYWORD=downing, 2001.
6. Microsoft Corporation, "Prepared text of remarks by Craig Mundie, Microsoft senior vice president: The commercial software model," www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp, 2001.
7. Paula Rooney, "Ballmer: Linux is top threat to Windows," www.techweb.com/wire/story/TWB20010110S0006, 2001.
8. "GNU General Public License," <http://www.gnu.org/copyleft/gpl.html>, 1991.
9. "GNU Lesser General Public License," <http://www.gnu.org/copyleft/lesser.html>, 1999.

第2章 系统体系结构

本章要点:

- 开拓者项目简介
- Silverjack 度假胜地的布局
- 开拓者项目的需求
- 开拓者项目的系统体系结构

服务于共同目标的一组设备组成的网络定义了一个系统。在最初的系统设计阶段，人们往往会创建一个称为系统体系结构的框架。系统体系结构在模块一级定义系统的组成部分以及它们之间的接口。系统体系结构为开发人员提供了对项目或产品在整体和结构上的理解，并在此范围内开展开发工作。系统体系结构提供的是系统的宏观布局。另外，一组高层次的系统需求有助于定义系统体系结构。系统体系结构不涉及组成它的各模块的技术细节。诸如组件规范和选择这样的细节是在模块的设计阶段决定的。

本章介绍开拓者 (Trailblazer) 项目——一个冬季度假胜地的自动化项目。该项目开始于高层次系统需求的形成，而后根据需求定义了包括各功能模块及其接口的系统体系结构。本章只阐述开拓者项目的系统体系结构，不涉及任何关于功能模块的技术细节。本书跟踪了开拓者项目使用嵌入式 Linux 的开发和功能模块的设计和实现。

2.1 开拓者项目简介

位于洛基山某处的 Silverjack 度假胜地的的发展十分迅速。世界各地的游客汇集到 Silverjack，在风景如画的场地享受各种冬季活动。由于有丰富的天然雪和人造雪，活动季节可以从 11 月份持续到下一年的 4 月份。Silverjack 的 10 部缆车使游客可以到达各种不同的地势，无论他们是滑雪者、滑雪板爱好者还是穿雪靴的旅行者。所有这些元素使 Silverjack 成为一个颇具吸引力的冬季度假场所。自从开业以来，其收入持续增长，董事会和投资者都非常高兴。有远见的度假胜地委员会知道，科技的应用可以使高山更加安全，运营更加高效，可以带给游客更加愉快的体验。他们知道最前沿的科技有助于将游客吸引到 Silverjack 来。

委员会为开拓者项目拨出了资金，该项目是一个服务于所有 Silverjack 区域的数据采集和控制系统。委员会希望这个信息系统是可靠、健壮和低成本的，要尽可能使用现成的硬件，而且要在 6 个月内完成设计、部署和测试。

Silverjack 工程部门从运营部门得到输入信息，在 3 个领域规划出一组高层次需求：安全、运营和供应商。这些需求不包含任何特定的技术细节，而是给出指导开发的方针。委员会审核并批准了这些高层次需求。他们准许工程部门启动开拓者项目。

工程部门以极大的热情开始了开拓者项目的设计。但他们很快发现，该项目基本上是一个开发周期很短的大型集成工程。工程部门需要一个灵活和可靠的解决方案，将串行、并行、输