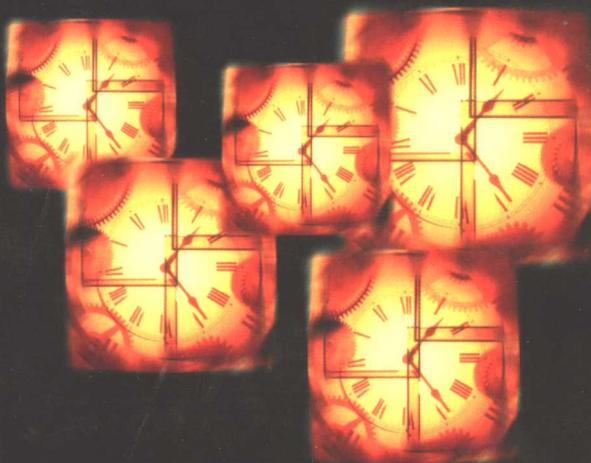


嵌入式系统软件教程

(美) David E. Simon 著 陈向群 等译

An Embedded Software Primer



David E. Simon

CD contains the real-time
operating kernel,
 $\mu\text{C}/\text{OS V1.11}$



An Embedded Software Primer



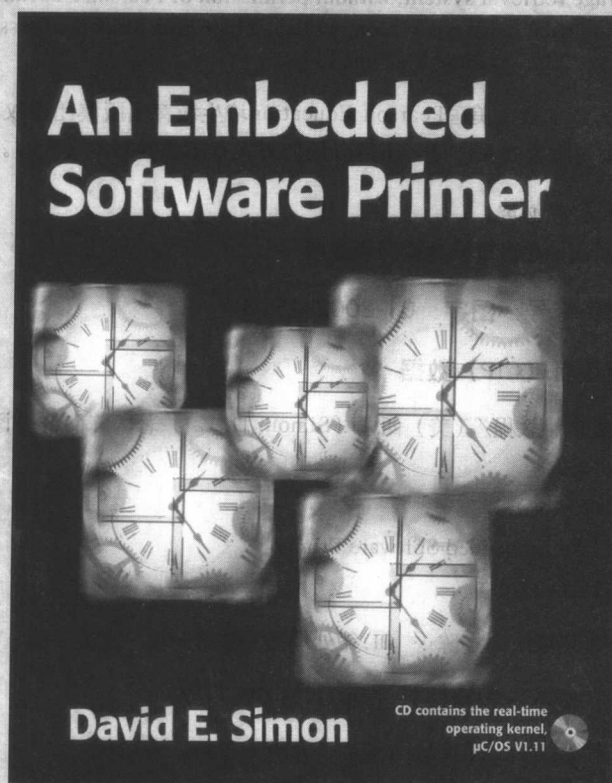
机械工业出版社
China Machine Press



计 算 机 科 学 丛 书

嵌入式系统软件教程

(美) David E. Simon 著 陈向群 等译



An Embedded Software Primer



机械工业出版社
China Machine Press

本书综合讨论嵌入式系统软件领域尤其是实时操作系统的各种技术。作者长期从事嵌入式系统软件设计和开发,把各种设计经验、开发技巧以及可能遇到问题的解决方案,有机地与嵌入式软件的基本原理和基础知识结合在一起。本书主要内容包括:软件技术的硬件基础、实时操作系统理论与设计、嵌入式软件开发工具、调试技术等。本书并不针对任何特定的微处理器或实时操作系统进行讲解,也不面向任何特定软件设计方法。

本书可作为高等院校的计算机、电子类相关专业嵌入式软件课程的教科书,也可作为嵌入式软件爱好者的自学读物。无论是有经验的软件工程师,还是学过编程课程的学生,都可以从本书中获益。

Authorized translation from the English Language edition entitled *An Embedded Software Primer* by David E. Simon, published by Pearson Education, Inc., publishing as Addison-Wesley (ISBN 0-201-61569-X), Copyright © 1999 by Addison-Wesley.

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

Chinese simplified language edition published by China Machine Press.

Copyright © 2005 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2002-4176

图书在版编目(CIP)数据

嵌入式系统软件教程/(美)西蒙(Simon, D. E.)著;陈向群等译.-北京:机械工业出版社,2005.9

(计算机科学丛书)

书名原文:An Embedded Software Primer

ISBN 7-111-16937-9

I.嵌… II.①西… ②陈… III.微处理器-系统开发-高等学校-教材 IV.TP332

中国版本图书馆CIP数据核字(2005)第079417号

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:姜淑欣

北京京北制版厂印刷·新华书店北京发行所发行

2005年9月第1版第1次印刷

787mm×1092mm 1/16·19.5印张

印数:0 001-4 000册

定价:39.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭集了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师提供服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证,但我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

电子邮件: hzjsj@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

前 言

本书为初学者系统讲解嵌入式软件基本原理。综合讨论了有关这个领域的各种技术，尤其讨论了对实时操作系统的恰当运用，而实时操作系统是许多嵌入式软件的基础。除了介绍这些系统的功能之外，本书还说明了如何有效地使用这些操作系统。

在阅读本书之前读者不需要对嵌入式系统的软件和有关问题有任何的了解，作者会从基础开始进行讨论。不论你是一位有一年或多年编程经验的软件工程师，还是刚刚学习了一些编程课程的学生，你都应该熟悉一些计算机程序设计的基本概念，应该理解编写程序过程中的有关问题。本书要求读者学习过C程序设计语言，因为C是嵌入式系统中的“国际通用语言”。如果读者计划进入嵌入式领域，那么迟早都得学习C语言。当然，通晓一些汇编语言的知识也会有帮助的。

你肯定看到过一些关于软件的书足有800页、900页，甚至1000页厚，也许你已经发现这本书则薄得多。这是作者的一种愿望，目的就是让读者尽可能多地翻阅它。这本书没有标榜“读者所要了解的有关嵌入式系统软件的一切”。没有人能够写出这样的书，即使某个人能够写出来，读者无论如何也不会读它的。本书更像是提供“想开始从事嵌入式系统软件工作所需要了解的”知识，告诉读者在遇到问题时所需要掌握的必要知识，并且让读者对所涉及的系统有所认识，从而能够解决所遇到的问题。

本书并不针对某一种特定的微处理器或实时操作系统，也不面向某一特定软件设计方法。无论是采用哪一种微处理器、哪一种实时操作系统、哪一种软件设计方法，原理都是相同的。本书中，我们专注于原理——可以运用到几乎所有的嵌入式系统项目的原理。在读者需要了解特定的微处理器以及实时操作系统时，可以参考硬件和软件销售商随产品提供的手册。本书会告诉读者应该查找哪些信息。

本书既不是学术性的也不是理论性的著作，而是为读者提供工程方面的信息以及相关建议。

总之，这本书是编写嵌入式系统软件所需要的一块基石。

致谢

如果没有别人的帮助，一个人就不会有足够好的创意来写这样一本书，或者也不会有足够的毅力坚持把它完成。所以，下面按照时间的先后顺序介绍所有帮助我完成这本书的人们。

首先要感谢Probitas公司的人们：感谢A. J. Nichols，他使得公司成为一个如此富有思想且高质量的软件环境；感谢Michael Grischy，感谢他正在进行的关于嵌入式系统设计的讨论和编码风格；感谢Richard Steinberg，他检查了本书的代码示例；感谢Ric Vilbig，他检查了关于硬件的两个章节，并改正了我的许多误解。

我的妻子，Lynn Gordon，鼓励我写这本书，并正确地预见到我将非常喜爱做这件事。感谢你鼓励我开始，感谢你所做的编辑工作……即使在英语用法的细节上，你也总是正确的。

感谢6个班的全体学生：感谢那些提出深刻问题，并因此促使我理清思路的学生们；感谢那些提出基本问题，并因此促使我详细阐明解释的学生们；另外，感谢所有拿到我的初稿后

保持沉默的学生们。

感谢Dragon Systems有限公司的聪明的伙计们，感谢你们编写的*NaturallySpeaking*程序，这个语音识别程序是如此之好，使得我不用受很多打字之苦就可以准备出手稿。

非常感谢Jean Labrosse允许我在本书中使用他的实时操作系统 μ C/OS。你写出这个系统，并允许它在教育领域免费使用，真是帮了世界一个大忙。

感谢John Keenan，你教会我许多硬件的知识，并且直率地指出我的手稿中的许多错误。

以下是检查了手稿的第一版并提出大量好的建议的人们，他们中许多人的名字我已经在书中提到：Antonio Bigazzi、Fred Clegg、David Cuka、Micheal Eager、John Kwan、Tom Lyons以及Steve Vinoski。感谢你们！

感谢Zilog公司的Mike Hulme，允许我在第3章的最后使用图表的例子，并提供了清晰的副本。

最后，感谢Debbie Lafferty和Jacquelyn Doucette监管了本书从开始到出版的所有阶段；感谢Ann Knight和Regina Knox指出所有不知为何我写的时候是清楚的但后来别人阅读时却不清楚的地方；感谢Laurel Muller将我画的草图制成清晰的图表。

关于本书和光盘

嵌入式系统的不同之处

嵌入式系统中一个非常令人遗憾的地方在于，有关术语不是很统一：每一个概念都有2~3个不同的词表述，每一个词又有4~5个略微不同的意思，而读者只能忍受这件事。在本书中，我们将指出这些意思的不同之处，之后我们会为每一个词指定一个明确的意义来帮助读者不会混淆。当读者阅读其他的书或是与人交谈时，读者需要注意他们的用词可能与在本书中这些词的意思有一些不同。

另一个令人遗憾的地方在于嵌入式系统这个词涵盖了很大范围的产品，概括起来是很困难的。嵌入式系统可建立在不同的微处理器上，从Z8（不能够使用外存的8位微处理器）到PowerPC（可以访问千兆字节的32位微处理器），其应用程序的代码大小也从500个字节到数百万个字节不等。

基于这些原因，以及应用的不同，在嵌入式系统这个领域里没有什么通用的。某个被85%的工程师视为最佳实践的规则，在另外15%的工程师眼里有时为了系统运转不得不被打破。本书将关注适用于85%情况下的规则，重点强调概念和规则背后的原理，并帮助读者决定是应该遵从这些通用规则呢，还是读者的项目属于另外15%的情况。

本书的章节间依赖关系

尽管这本书是打算让读者按顺序从头读到尾的，尽管每一章都至少依赖一点它前面的章节，但是如果读者喜欢，还是可以跳过一些章节。由于本书对每一部分都从最基本的知识讲起，如果读者已经知道了一些相关的知识，可以跳过这些段落。章节之间最主要的依赖关系如下页图所示。

例如，如果读者已经很熟悉硬件，或者不需要了解这些，读者可以跳过第2章和第3章。但是不要尝试在没有阅读第4章、且不了解相关知识的情况下就阅读第6章。

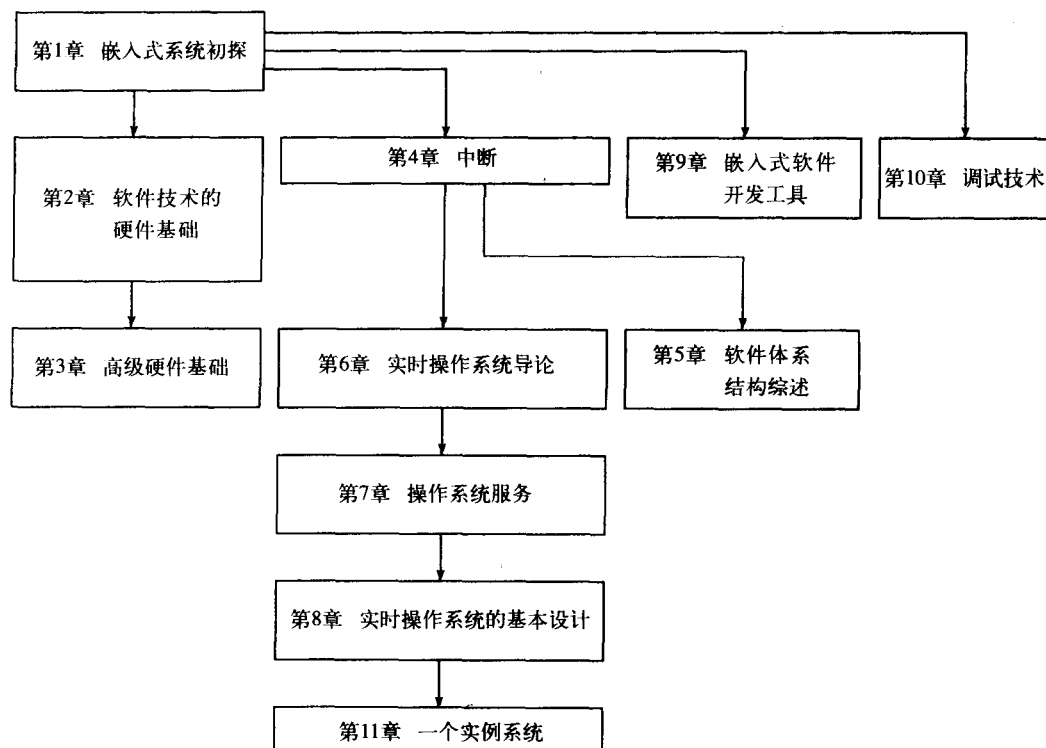
C++

尽管C++在嵌入式系统领域中是一个越来越流行的语言，但读者不会在本书中看到它。这并不是不鼓励读者使用这个在嵌入式系统领域和应用程序领域中都如此通用的C++语言。但是C++有一个公认的缺点在于：它是一种复杂的语言，在很多方面都比C语言复杂和难以捉摸。本书所讨论的编程原则对于C和C++都是通用的（也包括Ada、Java、BASIC……）。

C!!

在诸如本书这样的书中提供容易理解示例，常见的一个问题（也是一般软件的问题）是，关键之处总是隐藏在细节中。为了避免这种事情在本书中出现，书中的有些示例不是全部用C

而是用C! !写成的。



除了放了两个惊叹号之外，C! !和C完全一样，计算机完成惊叹号后所描述的一切。

例如：

```

if ( x != 0 )
{
    // 从硬件计时器中读出值
    // 进行必要的计算
    y = // 将计算结果存储在y中
}
if ( y > 197 )
{
    // 点亮报警灯
}

```

如果x不为零，此程序将从硬件计时器中读出值，而后将进行必要的计算并将结果存储在y中。如果y>197，此程序将点亮报警灯。

接下来假定只使用C! !的特殊性质来编写一些应用中依赖硬件或需要进行指定计算的代码。例子中的这些部分是用普通的C语言编写的^①。

① 如果所有的程序都能用C!!来编写是最好不过的了，但是C!!的编译器仍然处于开发阶段。

匈牙利变量命名方法

本书中很多代码实例使用匈牙利标记法来命名变量。下面简要介绍此规范。在匈牙利标记法中，变量名中包含关于它的类型信息。例如，整型（int）变量名以字母i开头，字节型（byte）变量以“by”开头等。而且，变量名的前缀可以表明此变量是一个指针（“p_”）还是一个数组（“a_”）等。下面是一些典型的变量名以及我们从这些变量名中可推断出的信息：

byError——字节型变量（可能包含一个错误代码）。

iTank——整型变量（可能包含油罐的数目）。

p_iTank——整型指针。

a_chPrint——字符型数组（大多数情况，信息将被打印出来）。

fDone——一个标记（表明一个过程是否完成）。

虽然一开始理解这些名字有点费力，但是人们普遍认为变量名中包含的少量信息可以方便编程，这就是匈牙利标记法流行的原因。^①

μC/OS

如果读者已经看了第6章中关于实时操作系统的讨论，读者可能想亲手试用这样一种系统。为了帮助读者实现这样的想法，本书附带的光盘会提供μC/OS（发音为“micro-see-oh-ess”）。此光盘上的内容已经得到μC/OS的作者Jean Labrosse的许可。读者可以将μC/OS作为一种学习工具来使用，但是不可以在未经许可的情况下用于任何商业产品。μC/OS不是共享软件，下面有关于μC/OS的许可。第6、7章和表11-2可以引导读者使用此系统。如果读者想了解关于此系统更多的信息，可以参见本书后“进一步阅读”书目列表中Labrosse的书籍，或者到μC/OS的网站了解更多信息。

μC/OS许可证

μC/OS的源代码和目标代码可以免费分发给可信赖的大学和学院的学生使用，只要不用于商业应用就不需要许可证。也就是说，将μC/OS用于教育目的并不需要许可证。

要将μC/OS嵌入在商业产品中，需要得到目标代码分发许可证，它允许将μC/OS用在盈利性产品中但需要支付一定量的许可费用，需要联系Labrosse先生来商讨价格。

如果打算分发μC/OS的源代码，同样需要得到源代码分发许可证，同样也需要支付许可费用，可联系Labrosse先生来商讨价格。

可以通过以下联系方式联系到Jean Labrosse先生：

Jean.Labrosse@uCOS-II.com

或者

Jean J. Labrosse

949 Crestview Circle

Weston, FL 33327

① 本书中用到的匈牙利标记法只是一个衍生版本，并不是原始版本。衍生版本与原始版本的主要区别在于：
(1) 原始版本并没有使用下划线将前缀与变量名的其他部分分开；(2) 衍生版本使用与变量命名相同的规范来命名函数。

USA

1-954-217-2036 (电话)

1-954-217-2037 (传真)

官方网站为www.ucos-ii.com。读者可以联系Labrosse先生获得关于 μ C/OS的支持,但前提是读者已经从 μ C/OS网站得到了 μ C/OS的最新版本和补丁。

附带光盘中的实例程序和 μ C/OS需要在DOS环境下使用Borland C/C++编译器方可使用。现在可以花\$49.95得到此编译器的教学版本(只用于学习目的,不能用于商用产品的开发)。不同的零售商都提供此编译器,读者也可以通过www.borland.com与Borland公司联系。

目 录

出版者的话	
专家指导委员会	
前言	
关于本书和光盘	
第1章 嵌入式系统初探	1
1.1 嵌入式系统的举例	1
1.2 典型的硬件	5
本章总结	6
第2章 软件技术的硬件基础	7
2.1 术语	7
2.2 门	9
2.3 其他一些基础知识	11
2.4 时序图	16
2.5 内存	18
本章总结	22
习题	23
第3章 高级硬件基础	25
3.1 微处理器	25
3.2 总线	26
3.3 直接内存访问	32
3.4 中断	35
3.5 其他常用元器件	35
3.6 与微处理器集成的元器件	41
3.7 示意图规范	44
3.8 示意图实例	44
3.9 有关硬件的最后几句话	46
本章总结	46
习题	46
第4章 中断	49
4.1 微处理器体系结构	49
4.2 中断基础知识	51
4.3 共享数据问题	55
4.4 中断延迟	63
本章总结	68
习题	69
第5章 软件体系结构综述	71
5.1 轮转结构	71
5.2 带有中断的轮转结构	73
5.3 函数队列调度结构	79
5.4 实时操作系统结构	81
5.5 软件结构的选择	82
本章总结	83
习题	83
第6章 实时操作系统导论	87
6.1 任务和任务状态	88
6.2 任务和数据	91
6.3 信号量和共享数据	98
本章总结	108
习题	108
第7章 操作系统服务	111
7.1 消息队列、信箱和管道	111
7.2 定时器函数	119
7.3 事件	124
7.4 存储管理	127
7.5 RTOS环境中的中断程序	129
本章总结	134
习题	134
第8章 实时操作系统的基本设计	141
8.1 概述	141
8.2 原理	142
8.3 举例	153
8.4 信号量和队列的封装	160
8.5 硬实时系统调度问题	167
8.6 节省存储空间	167
8.7 节省能源	170
本章总结	171
习题	172
第9章 嵌入式软件开发工具	173
9.1 宿主机和目标机	173

9.2 嵌入式系统的链接器/定位器	174	第11章 一个实例系统	219
9.3 把嵌入式软件加入到目标系统中	184	11.1 程序是做什么用的	219
本章总结	187	11.2 程序运行的环境	221
第10章 调试技术	189	11.3 源代码指南	223
10.1 在宿主主机上进行测试	189	11.4 源代码	225
10.2 指令集模拟器	202	本章总结	276
10.3 assert宏	203	习题	277
10.4 使用库工具	205	后记	279
本章总结	217	进一步阅读	281
习题	218	索引	283

第1章 嵌入式系统初探

现如今，随着微处理器的体积变得越来越小、价位越来越低，越来越多的产品中“嵌入”了微处理器，以实现其“智能化”。这些微处理器芯片以及相关软件已然应用到许多方面，如数码相机、数字手表、电梯、汽车发动机、温控装置、工业控制设备、科学仪器与医疗器械等。人们以“嵌入式系统”这个名词来表示已经融合到各种产品中的计算机系统。

嵌入式系统的软件必须能解决一些在台式机或大型计算机应用软件中不存在的问题。嵌入式系统经常要同时完成若干任务，所以必须能够及时响应外部事件（比如，有人按下一个电梯按钮），必须能在无人干预的情形下应对所有的异常情况。因而，它们的工作有着严格的时间限制。

1.1 嵌入式系统的举例

我们通过讨论几个实例，来帮助读者加深对嵌入式系统软件的理解，这样可以把问题具体化。随着对一些问题和有关解决方案讨论的进一步深入，我们会经常返回来研究这些实例。

1.1.1 电报机

首先讨论的是以“电报机”命名的一个系统，其名称源于历史发展。电报机可与只有高速串行网络接口的打印机相连。在外观上，电报机就像一个小塑料盒子，2~3英寸宽，约半英寸厚；一侧有电缆插入打印机串行接口，另一侧的连接器插入到网络中。电报机的外观如图1-1所示^①。

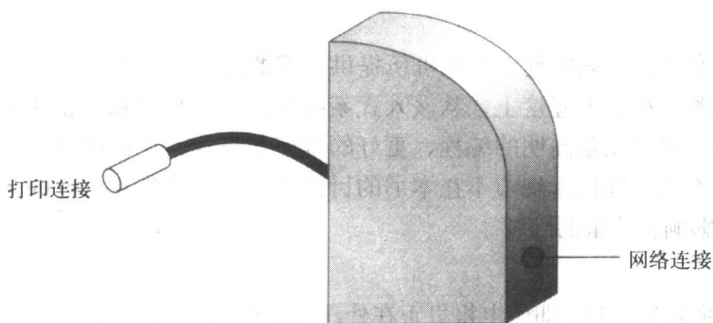


图1-1 电报机

显然，电报机必须从网络接收数据，然后将数据复制到串口上，但实际上，电报机要完成的工作要比这些复杂得多。下面是电报机必须完成的一些工作：

- 在网络上，数据有时并非有序到达，甚至有的数据在传输途中丢失，而有些则到达了

① 电报机必须与苹果喷墨打印机一起工作，其外形决定了它可直接连到打印的背面一侧。电报机还有一个串口，可直接连到Macintosh计算机上，不同的电报机款式可以应用于不同的计算机网络。

完全相同的两份。电报机必须对网络上这些混乱的数据加以整理，并为打印机提供清晰的数据流。

- 网络上通常连有许多计算机，这些计算机有可能同时发出打印请求。打印机一般仅能与一台计算机连接，这样电报机只能一次向打印机提出一个打印作业，甚至有时设法拖延其他计算机的打印作业。
- 网络打印机必须向网络上发出请求的计算机报告其工作状态，即使它当前正在为其他计算机打印作业。传统的串口打印机本身无法实现这一点，但电报机必须能做到。
- 电报机必须能在无客户配置的情况下，与各种不同的打印机协同工作，并能够判断出与之连接的打印机类型。
- 对于特定的事件，电报机必须能够迅速做出响应。例如，对传给电报机的各种网络帧，电报机必须要在200 μ s内发出响应。
- 电报机必须维持对时间的记录。例如，如果一台正发送给电报机打印数据的计算机崩溃了，电报机必须最后放弃该打印作业（可能在2分钟以后），并通过网络上另一台计算机取出数据并打印。否则，一台计算机的崩溃将造成打印机的所有使用者都不能继续使用打印机。

2

1.1.2 电报机的发展与面临的挑战

为了满足上述需求，电报机内嵌了微处理器。这样，电报机的软件覆盖面更广阔也更复杂，较其外观可能给人留下更深的印象。在开发这些软件过程中会遇到哪些问题？在继续往下读之前，读者不妨先自己考虑一下可能遇到的问题，并把它们记录下来。

当然，首先要求电报机的软件是逻辑正确的，不能丢失某台计算机的请求，不能漏掉数据或者错报状态。这也是对所有软件在嵌入式领域和应用领域的要求。

而编写这样一个电报机软件（类似于许多其他嵌入式系统的软件）还会面临一些额外的挑战，下面我们就来逐一讨论。

1. 吞吐量

打印机的打印速度最快也只能赶上电报机所提供数据的速度。电报机不能成为联网计算机与打印机之间的瓶颈。在很大程度上，从嵌入式系统得到更多数据和使软件运行得更快是相似的问题。其解决方案可以是高明的编程、更好的排序和查找、更好的算法，或者更易解析的数据结构等等。不过这些技术细节不在本书的讨论范围内，我们要讨论的是一些实时操作系统中可能出现的影响吞吐量的问题。

2. 响应

当一个关键的网络帧到达时，即使电报机正在处理别的任务，也必须在200 μ s内做出响应。软件的编制则要确保这一点。我们将对响应展开讨论，因为这是嵌入式系统的普遍问题，而且所有的解决方法都意味着需要各种权衡。

人们经常使用一个比较模糊的词——“速度”，然而嵌入式系统的设计者们要考虑的是两个分立的问题——响应和吞吐量，解决的手段也各不相同。事实上，解决好其中一个问题通常会使另一个变得更糟糕。所以本书中将分别坚持使用“响应”和“吞吐量”这两个术语，而避免使用“速度”一词。

3. 易测性

判断电报机能否正常工作并不容易，问题在于它的许多软件要处理一些异常事件。从这个

3

角度讲,电报机是一类典型的嵌入式系统,因为它要在无人干预的条件下处理好一切,例如,系统大量的代码用于解决丢失的数据问题。然而数据又并不是那么容易丢失,尤其在实验中,网络建立的非常完善,各部件一律全新,连线皆为15英尺长,这使得那些程序很难被检测。

类似地,电报机要处理并发事件。例如,如果两台计算机几乎同时给出打印任务,软件能恰当处理吗?当然,嵌入式系统中有代码来处理这种情况,但是,怎样才能让两台计算机几乎同时给出打印任务来测试这段代码呢?

我们将就此作一些讨论。

4. 可调试性

当电报机软件中的一个错误被检测出来的时候,会发生什么情况呢?电报机没有显示屏,没有键盘,没有扬声器,甚至没有指示灯。当一个错误出现,你得不到任何标识或者对话框,电报机就停止工作了!是网络的软件错误?是记录计算机打印任务的软件错误?是报告打印机状态的软件错误?电报机就这样停止工作了,而没有任何提示。

不幸的是,电报机的停止运行不能带来任何关于错误的信息。更糟糕的是,在没有显示屏、没有键盘的情况下无法运行任何调试程序!设计者必须通过其他方法来确定发生了什么故障。我们将讨论一些用于嵌入式系统软件检错的方法,以及一些能避免危害到整个系统的严重错误的方法。

4

5. 可靠性

如同大多数的嵌入式系统一样,电报机是不允许崩溃的。尽管用户似乎能够容忍在台式机上不时地重新启动系统,但是没人会对一个崩溃的小塑料盒子有耐心的。一个尤其糟糕的情况是,应用软件可以通过在显示屏上发消息来询问用户该做些什么。但是嵌入式系统软件则没有这种可能,无论发生什么,嵌入式系统软件都必须在无人干预下运行。

6. 内存空间

电报机只有很有限的内存,其中32KB用于程序,32KB用于数据。受成本的制约,内存就只能这么大了。即使内存价格便宜了,但是仍然不是免费的。所以,使系统适应有限的内存空间是设计嵌入式系统软件的工程师必备的技能,这将在下面讨论。

7. 程序安装

电报机软件的安装是不能通过鼠标单击图标来完成的。下面将会讨论在嵌入式系统中安装软件所需要的特殊工具。

1.1.3 无线条形码扫描仪

再来看一个嵌入式系统的例子——无线条形码扫描仪。使用者一按下开关,无线条形码扫描仪就会发射激光读取条形码,再通过无线通信部件传给收款机(见图1-2)。

无线条形码扫描仪软件开发的问题与电报机的相比较有何不同呢?

事实上,大体是相同的。无线条形码扫描仪不存在吞吐量问题。条形码没有大量数据,使用者也不用飞快地按开关。另一方面,无线条形码扫描仪还有一个电报机所没有的问题。

5

能耗

因为扫描仪是无线的,电池就是惟一的能源来源。又因为扫描仪是手提式的,电池的重量被限制在普通用户能够舒适握持的水平。用户希望电池能够维持多长时间?显然,答案是:永远,当然这是不可能实现的。那么次优的答案是什么?