

嵌入式系统经典丛书

Real-Time Embedded Systems
Design Principles and Engineering Practices

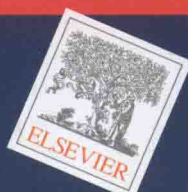


实时嵌入式系统 设计原则与工程实践

[美] Xiacong Fan

林 赐

著
译



清华大学出版社

嵌入式系统经典丛书

实时嵌入式系统 设计原则与工程实践

[美] Xiaocong Fan 著
林赐 译

清华大学出版社

北 京

Xiaocong Fan

Real-Time Embedded Systems: Design Principles and Engineering Practices

EISBN: 978-0-12-801507-0

Copyright © 2015 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by Elsevier (Singapore) Pte Ltd. and Tsinghua University Press.

Copyright © 2016 by Elsevier (Singapore) Pte Ltd and Tsinghua University Press. All rights reserved.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)出版与发行。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

北京市版权局著作权合同登记号 图字: 01-2016-7642

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

实时嵌入式系统 设计原则与工程实践 / (美)樊晓聪(Xiaocong Fan) 著, 林赐 译. —北京: 清华大学出版社, 2017

(嵌入式系统经典丛书)

书名原文: Real-Time Embedded Systems: Design Principles and Engineering Practices

ISBN 978-7-302-45559-2

I. ①实… II. ①樊… ②林… III. ①微型计算机—软件设计 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2016)第 277367 号

责任编辑: 王 军 于 平

封面设计: 牛艳敏

版式设计: 思创景点

责任校对: 曹 阳

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 34.25 字 数: 834 千字

版 次: 2017 年 1 月第 1 版 印 次: 2017 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 79.80 元

产品编号: 069042-01

译者序

岁月荏苒，斗转星移。自从第三次科技革命以来，计算机技术在美国首先兴起，在短短的几十年间迅速席卷了整个世界。互联网的出现，更加强了世界之间的联系，人们得以共享信息资源。正当电子商务、4G 网络、人工智能技术此起彼伏，方兴未艾，物联网和工业 4.0 的时代就已经出现在我们的面前。嵌入式系统技术是物联网和工业 4.0 的关键技术。它综合了计算机软硬件、传感器技术、集成电路技术、电子应用技术，广泛地应用在生产生活的各个领域，小到人们身边的 MP3，大到航天航空的卫星系统。有人把物联网比作人体，传感器相当于人的眼睛、鼻子、皮肤等感官，网络是用来传递信息的神经系统，而嵌入式系统则是人的大脑，在接收到信息后进行分类处理。

本书的出现恰逢其时，它集合了来自实时系统、嵌入式系统和软件工程的各種新概念和新主题，为我们描述了嵌入式系统应用程序软件开发的整个过程。通过本书，读者可以了解到嵌入式系统不同体系架构的利弊、POSIX 实时扩展，以及开发符合 POSIX 标准的实时应用程序、跨平台开发的概念和所面临的挑战、多任务设计和任务间通信技术（共享内存对象、消息队列、管道、信号）、如何使用内核对象（如信号量、互斥体、条件变量）来解决 RTOS 应用资源共享的问题等。

路漫漫其修远兮，吾将上下而求索。关于实时嵌入式系统的书籍、文献、资料汗牛充栋，卷帙浩繁，本书作者兢兢业业，博览群书，去粗取精，在书中使用最先进的算法和产业标准，终于付梓。知易行难，实时嵌入式系统科学是一门实践性很强的学科，没有脚踏实地的精神和温故知新的态度，难以掌握知识的精髓，因此建议读者在学习完每一章后，亲自完成章节后的作业，在机器上敲一敲代码，跑一跑程序，如此才能登堂入室，敲开嵌入式系统的大门。

冰冻三尺，非一日之寒。技术的掌握并非一朝一夕之功。因此，选择了学习实时嵌入式系统，准备成为一名嵌入式系统工程师，就要有打“持久战”的思想准备，持之以恒地不断学习新技术、新概念和新知识，唯有这样，才能水滴石穿，不落后于时代，成为社会的中流砥柱，引领技术的潮流。中国的历史长河绵延千里，在科技上曾经繁荣一时，留下了一笔又一笔宝贵的知识遗产，如《本草纲目》、《齐民要术》、《梦溪笔谈》、《天工开物》等。但是到了近代，中国却落后了。第四次科技革命已然到来，这是一个机遇，也是一个挑战，面对这个日新月异的世纪，中国已经做好了准备，但是读者做好迎接危机的准备了吗？

在这里要感谢清华大学出版社的编辑们，她们为本书的翻译投入了巨大的热情并付出了很多心血。没有她们的耐心和帮助，本书不可能顺利定稿出版。

译者才疏学浅，唯恐误人子弟，在这本经典之作的翻译过程中，如临深渊，如履薄冰，

力求达到前辈提出的“信、达、雅”的标准。但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。感激不尽！本书全部章节由林赐翻译。

译 者

于加拿大渥太华大学荔枝街

致 谢

首先，我衷心地感谢能有这个机会，向许多优秀且可用的实时系统或嵌入式系统教材学习，这些教材的作者包括：W.S. Liu、David E. Simon、Rob Williams、Qing Li 和 Bruce P. Douglass，这里仅提到了一些作者。在书中的一些章节，他们可以直接或间接地认识到自己带来的影响。

我要感谢审稿人员众多有见地的意见和启发性的建议。我要感谢我的学生，在我思考如何最好地组织和讲授实时嵌入式系统课程时，他们给予我很大的影响。

最后的手稿由 SPi 全面编辑。其中，最特别感谢的是 Elsevier 的 Tim Pitts、Charlie Kent 和 Nicky Carter。没有他们耐心、妥善地处理，本书不可能与读者见面。

前 言

嵌入式系统是在一个较大的系统中设计以执行专用功能的电子系统。实时系统为在临界事件的最坏情况下的响应时间提供了保证，同时为非临界事件提供了可接受的一般情况下的响应时间。当实时系统设计成为一个嵌入式组件时，这个组件就称为实时嵌入式系统，此系统广泛地应用在消费、工业、医疗和军事方面。

随着我们的日常生活越来越依赖于嵌入式技术，近几年来，对具有开发实时嵌入式软件技能集的工程师的需求大幅增长。结果是，培养学生在嵌入式软件的设计和实现方面的能力变得日益重要。本书专门为高年级本科生或攻读软件工程、计算机工程或相关学科的研究生撰写，但也有益于专注嵌入式软件开发的在职工程师。

本书采用了协同合作的方法，引进了实时系统、嵌入式系统和软件开发原则的概念和主题。读者不仅可以深入了解有关微处理器、中断和跨平台开发流程的概念，意识到实时建模和调度的重要性，并且可以得到良好的软件工程实践的训练，这些实践包括建模文档、建模分析、设计模式和系统标准的一致性。

本书介绍开发实时嵌入式软件至关重要的三个方面。

首先，开发实时嵌入式系统软件涉及许多活动，包括需求规格说明、时序分析、体系架构设计、多任务设计以及跨平台的测试和调试。本书涵盖嵌入式软件开发的全过程，对一些主题进行全面充分的解释，而对另外一些主题则一笔带过(例如，调试和测试)。

具体来说，本书系统地介绍各种嵌入式软件架构，专注于实时操作系统，这是在大型实时操作系统中采用的最先进的架构。此外，我们选择性地强调可重复使用的设计解决方案。如表 0.1 所示，本书介绍许多设计模式，这些设计模式代表了在广泛的实时嵌入式系统中可重用的最佳实践。

表 0.1 设计模式总结

类 别	模 式 名	书 中 位 置
ISR	ISR-Pattern-min	第 4.5.1 节
	ISR-Pattern-server	第 4.5.2 节
	中断链	第 4.5.3 节中的图 4.7
	中断级联	第 4.5.4 节中的图 4.9
	中断禁用	第 4.5.5 节中的图 4.11
	双缓冲	第 4.5.5 节中的图 4.12
	兑现第一个请求	第 12.3.2 节中的图 12.17
子类化	抽象-发生	第 6.3.3 节中的图 6.25
	泛化层次	第 6.3.4 节中的图 6.27

(续表)

类别	模式名	书中位置
软件架构	轮询 DAS	第 12.2.2 节中的图 12.10
	带有中断的轮询	第 12.3.2 节中的图 12.16
	FIFO 队列	第 12.4.1 节中的图 12.20
	优先级队列	第 12.4.2 节中的图 12.21
	串行端口设计模式	第 14.2.2 节中的图 14.5
静态任务调度器	基于时钟	第 15.2 节
	基于帧	第 15.3 节
	计时轮	第 22.3 节
信号量/互斥体	会合同步模式	第 18.3.1 节中的图 18.8
	多实例资源保护模式	第 18.4.1 节中的图 18.19
条件变量	栅栏同步模式	第 18.5.1 节中的图 18.24
	生产者-消费者模式	第 18.5.2 节中的图 18.27
	读-写锁模式	第 18.5.3 节中的图 18.30
消息队列	单向排队模式	第 19.3.1 节中的图 19.5
	确认-单向排队模式	第 19.3.2 节中的图 19.6
	双向排队模式	第 19.3.3 节中的图 19.7
	客户端-服务器排队模式	第 19.3.4 节中的图 19.10
管道	单向管道模式	第 20.3 节中的图 20.4
	双向管道模式	第 20.3 节中的图 20.4
死锁避免	分层消息传递模式	第 21.7.3 节中的图 21.8

DAS, 检测-确认-服务; FIFO, 先进先出; ISR, 中断服务例程

其次, 统一建模语言(Unified Modeling Language, UML)是用于指定、可视化、构造和文档化软件系统的图形语言。在各种工程问题中, 从单进程的嵌入式系统和单机用户应用程序到并发分布式系统, UML 都发挥着重大的作用。本书详细描述编写本书时最新的 UML 标准: UML 2.4。整本书中, UML 图同时用于系统设计和概念说明。特别是, 我们详细介绍了 UML 实时配置文件, 这样学生可以学习如何以专业的方式记录他们的实时系统设计。

另外, POSIX(即“可移植操作系统接口”)是开放性的操作系统接口标准, 在各种 Unix 操作系统变体之间用于开发互操作性和可移植性好的应用程序。构建在实时操作系统上的软件系统可以很容易地移植到其他 POSIX 兼容的操作系统上。

本书使用的是 POSIX.1-2008(2013 年版)。本书中涵盖的操作系统实现和概念与 POSIX.1-2008 标准完全兼容。书中提供的示例代码已在工业界广泛采用的操作系统 QNX 中测试。由于 QNX 是 POSIX 兼容的, 因此在不改变源代码的情况下, 也可以在另一个 POSIX 兼容的操作系统中编译并执行程序。

简单而言, 本书由以下 4 部分组成:

- 第 I 部分专门介绍实时嵌入式系统和迭代开发过程的基础知识。虽然我们重点放

在软件方面，但完全与底层硬件隔离既不可行、也不可取。由于这个原因，这个部分也包含微处理器和中断两个章节——这是希望构建嵌入式系统软件工程师的基本主题。

- 第II部分专门讲述实时系统的建模技术。具体来说，我们介绍UML所涵盖的建模工具——UML是在学术界和软件工业中普遍接受的一个标准。此外，我们介绍实时UML——指定系统模型中与实时相关约束的配置文件。全书一致地使用了UML图来说明关键的概念和设计模式。
- 第III部分专门讨论实时嵌入式系统的软件架构设计。我们开始于通用架构，这带领我们来到了最复杂的架构——实时操作系统。然后，我们的焦点切换到多任务和实时调度——这是每一个实时嵌入式系统设计师都要解决的两个关键问题。
- 第IV部分专门讲解系统实施。我们特别专注于在POSIX兼容的操作系统中可用的机制；这个方法意味着本书中给出的设计/实施模式也适用于其他POSIX兼容的操作系统。

这4个部分共有23个章节。在一个学期的课程中，教师可以为学生讲授感兴趣的部分章节。例如，在第3章中，为了在一个或两个课时中完成授课，可以跳过一些微处理器类型。如果在软件工程原则的预修课程中学过UML基础建模概念，那么第6章、第7章和第8章可用于自我阅读的任务或仅用作参考。根据不同的学生对操作系统基本概念的熟悉程度，在第IV部分中提到的一些主题(例如，消息队列、管道和信号)，教师可以因材施教。

为了帮助教师和学生使用这本书，在Elsevier指南网站<http://booksite.elsevier.com/9780128015070>中，我们提供了补充包，其中包括PPT讲义和源代码。扫封底的二维码也可以下载PPT讲义和源代码。

在本书中，未能涵盖实时嵌入式系统的每一个重大主题。在决定哪些主题适合软件工程师、哪些需要重点强调、哪些需要省略方面，这已是我能做出的最好判断。对于不同的观点，确实感兴趣的读者可以参考其他课本。

鼓励和欢迎来自同行的意见。请随时发送建议到Xiacong Fan, Behrend College, Pennsylvania State University, Erie, PA 16563, USA (e-mail: xfan@psu.edu)。我期待听到读者使用本书的反馈。

缩 略 语

AIC	advanced interrupt controller, 先进的中断控制器
ANSI	american National Standards Institute, 美国国家标准协会
BSP	board support package, 板级支持包
CAN	controller area network, 控制器局域网
CISC	complex instruction set computing, 复杂指令集计算
COFF	Common Object File Format, 公共目标文件格式
CPU	central processing unit, 中央处理单元
CPSR	current program status register, 当前程序状态寄存器
DAS	detect-acknowledge-service, 检测-确认-服务
DMA	direct memory access, 直接存储器访问
DSP	digital signal processor, 数字信号处理器
DTCM	data tightly coupled memory, 数据紧密耦合内存
EDF	earliest deadline first, 最早截止时间优先
EEPROM	electrically erasable programmable read-only memory, 电可擦除可编程只读存储器
ELF	Executable and Linking Format, 可执行与可链接格式
EOI	end of interrupt, 中断结束
EPROM	erasable programmable read-only memory, 可擦除可编程只读存储器
FIFO	first in first out, 先进先出
GPR	general-purpose register, 通用寄存器
GRM	General Resource Modeling, 通用资源建模
GRM	graphical user interface, 图形用户界面
HLP	highest locker protocol, 最高锁协议
HNERT	highest-priority nonempty ready-thread list, 最高优先级非空就绪线程列表
I2C	Inter-Integrated Circuit, 内部集成电路
IC	integrated circuit, 集成电路
ICE	in-circuit emulator, 在线仿真器
ICR	interrupt command/status register, 中断命令/状态寄存器
IEEE	Institute of Electrical and Electronics Engineers, 电气和电子工程师协会
IMR	interrupt mask register, 中断屏蔽寄存器
IP	Internet Protocol, 互联网协议
IRR	interrupt request register, 中断请求寄存器
ISR	interrupt service routine, 中断服务例程

LNA	low-noise amplifier, 低噪声放大器
LSb	least significant bit, 最低有效位
LSB	least significant byte, 最低有效字节
MMU	memory management unit, 内存管理单元
MOF	Meta Object Facility, 元对象机制
MSB	most significant byte, 最高有效字节
NVM	nonvolatile memory, 非易失性存储器
OCL	Object Constraint Language, 对象约束语言
OMG	Object Management Group, 对象管理组
OOM	object-oriented modeling, 面向对象建模
OOP	object-oriented programming, 面向对象编程
OS	operating system, 操作系统
PC	personal computer, 个人计算机
PCP	priority ceiling protocol, 优先级天花板协议
PCR	program counter register, 程序计数器寄存器
PDF	probability distribution function, 概率分布函数
PIC	programmable interrupt controller, 可编程中断控制器
PIT	programmable interval timer, 可编程间隔计时器
POSIX	Portable Operating System Interface, 可移植操作系统接口
PRF	pulse repetition frequency, 脉冲重复频率
QoS	Quality of Service, 服务质量
RAM	random-access memory, 随机存取存储器
RF	radio frequency, 射频
RISC	reduced instruction set computing, 精简指令集计算
RMA	rate-monotonic assignment, 单调速率分配
ROM	read-only memory, 只读存储器
RTOS	real-time operating system, 实时操作系统
RT-UML	Real-time Unified Modeling Language, 实时统一建模语言
SFR	special-function register, 特殊功能寄存器
SPI	serial peripheral interface, 串行外设接口
SRAM	static read-access memory, 静态只读存储器
TCM	tightly coupled memory, 紧耦合存储器
UART	universal asynchronous receiver-transmitter, 通用异步收发器
UML	Unified Modeling Language, 统一建模语言
UTC	Coordinated Universal Time, 协调世界时

目 录

第 I 部分 引 言

第 1 章 嵌入式系统和实时系统导论	1
1.1 嵌入式系统	3
1.2 实时系统	4
1.2.1 软实时系统	4
1.2.2 硬实时系统	5
1.2.3 实时系统的范围	6
1.3 案例分析：雷达系统	7
1.4 习题	10
第 2 章 跨平台开发	11
2.1 跨平台开发流程	11
2.2 硬件架构	12
2.3 软件开发	13
2.3.1 软件设计	13
2.3.2 系统编程语言 C/C++	13
2.3.3 测试硬件无关模块	18
2.4 生成目标映像	18
2.4.1 交叉开发工具链	18
2.4.2 可执行和链接格式	21
2.4.3 内存映射	26
2.4.4 案例研究：制作 QNX 映像	28
2.5 转移可执行文件对象到目标平台	29
2.6 在目标系统上的集成测试	30
2.7 系统生产	30
2.8 习题	31
第 3 章 微处理器入门	33
3.1 微处理器简介	33
3.1.1 常用的微处理器	33
3.1.2 微处理器特性	35
3.2 PIC18F8720 微型芯片	38

3.2.1 存储器组织	39
3.2.2 字写模式	41
3.2.3 字节选择模式	43
3.2.4 字节写模式	45
3.3 Intel 8086	46
3.3.1 存储器组织	47
3.3.2 独立的 I/O 地址空间	48
3.3.3 存储器地址空间	50
3.3.4 等待状态	52
3.4 Intel Pentium	53
3.4.1 总线状态转换	56
3.4.2 存储器组织	60
3.5 ARM926EJ-S	61
3.6 习题	64
第 4 章 中断	67
4.1 中断导论	67
4.2 外部中断	67
4.2.1 非向量中断	68
4.2.2 PIC 和向量中断	69
4.3 软件中断	74
4.4 内部中断	75
4.5 ISR 的设计模式	75
4.5.1 一般的 ISR 设计模式	75
4.5.2 具有服务器任务的 ISR	76
4.5.3 ISR 链	77
4.5.4 ISR 级联	78
4.5.5 与 ISR 进行数据共享	79
4.6 中断响应时间	81
4.7 案例分析：x86	82
4.7.1 硬件中断	84
4.7.2 综合汇总	85
4.8 案例研究：ARM 处理器	86

4.8.1 硬件中断	87	7.4 子系统建模	150
4.8.2 综合汇总	90	7.5 完整系统建模	153
4.9 习题	91	7.6 部署图	154
第 5 章 嵌入式系统的引导过程	93	7.7 习题	155
5.1 系统引导加载程序	93	第 8 章 UML 行为建模的基础	157
5.2 系统的引导过程	94	8.1 用例图和用例建模	157
5.2.1 加载嵌入式软件	94	8.1.1 用例图	157
5.2.2 准备执行嵌入式软件	95	8.1.2 用例描述	160
5.3 案例研究: AT91SAM9G45 引导过程	96	8.1.3 用例层次	161
5.4 加载嵌入在操作系统映像中的 ELF 目标文件	96	8.2 序列图	162
5.5 案例研究: 基于 QNX 嵌入式 系统的引导过程	97	8.3 活动图	167
5.6 习题	99	8.4 习题	171
第 II 部分 实时系统建模		第 9 章 UML 中有状态行为的建模	173
第 6 章 UML 结构建模基础	103	9.1 状态机图的基础	173
6.1 统一建模语言	103	9.1.1 状态	173
6.2 类图和类建模	104	9.1.2 转移和事件	174
6.2.1 类	106	9.1.3 伪状态	175
6.2.2 实例层次的关系	109	9.1.4 用状态机对网络协议建模	176
6.2.3 依赖关系	121	9.2 复合状态	178
6.2.4 泛化关系	125	9.2.1 进入点、退出点与历史	179
6.3 类建模原则	128	9.2.2 并发	180
6.3.1 模型演化	128	9.3 状态行为的继承	181
6.3.2 子类化	130	9.4 有状态对象时序图	183
6.3.3 最小信息冗余	131	9.5 举例: 雷达系统有状态行为 的建模	184
6.3.4 重构	134	9.5.1 收发机建模	185
6.4 对象图	137	9.5.2 链路驱动器的建模	186
6.5 包图	138	9.5.3 指挥消息器的建模	188
6.5.1 包引入	138	9.6 习题	189
6.5.2 包合并	139	第 10 章 实时 UML: 通用资源 建模	191
6.6 习题	141	10.1 实时 UML 配置文件	191
第 7 章 UML 体系架构建模	143	10.2 资源建模	194
7.1 体系架构的抽象层次	143	10.2.1 UML 核心资源模型	195
7.2 UML 结构图	144	10.2.2 动作和动作执行	196
7.3 建模组件	148	10.2.3 受保护资源的 UML 构 造型	197

10.2.4	资源使用	199	12.2.3	最坏情况下的事件响应 时间	253
10.2.5	资源-客户图	200	12.3	带有中断的轮询	254
10.3	时间建模	201	12.3.1	案例研究: 西蒙游戏	254
10.3.1	时间观念	201	12.3.2	通用架构	257
10.3.2	定时装置	202	12.3.3	最坏情况下的事件响应 时间	259
10.3.3	时间建模构造	204	12.4	基于队列的架构	260
10.4	并发建模	206	12.4.1	非抢占式 FIFO 队列	261
10.5	习题	209	12.4.2	非抢占式优先队列	262
第 11 章	实时 UML: 模型分析	211	12.5	习题	264
11.1	时序约束的启发	211	第 13 章	POSIX 和 RTOS	267
11.2	RT-UML 配置文件可调度性 建模子配置文件	214	13.1	POSIX 简介	267
11.2.1	RT-UML 配置文件中的可 调度性分析的元概念	214	13.1.1	POSIX 进程和线程	268
11.2.2	可调度性构造	219	13.1.2	POSIX 实时扩展	269
11.2.3	使用可调度性子配置 文件	221	13.1.3	POSIX 的兼容性和一 致性	275
11.3	RT-UML 配置文件性能建模 子配置文件	223	13.2	任务的静态结构和动态 行为	276
11.3.1	RT-UML 配置文件中的性 能分析的元概念	223	13.2.1	一般任务结构	276
11.3.2	性能构造	225	13.2.2	任务状态转移	278
11.3.3	使用性能子配置文件	227	13.3	实时操作系统	279
11.4	习题	231	13.4	POSIX 实时调度策略	282
			13.4.1	FIFO 调度策略	282
			13.4.2	轮询调度策略	283
			13.4.3	偶发服务器调度策略	284
			13.5	其他实时调度策略	286
			13.5.1	最小松弛度优先	286
			13.5.2	最早截止期限优先	287
			13.5.3	截止时间单调分配的 调度	288
			13.5.4	速率单调分配的调度	288
			13.6	习题	289
			第 14 章	多任务	291
			14.1	多任务简介	291
			14.2	多任务设计	292
			14.2.1	任务标识	292
			14.2.2	任务转换	295

第 III 部分 实时系统设计

第 12 章 实时嵌入式系统软件

架构

12.1	实时任务	237
12.1.1	最坏情况下的任务执行 时间	238
12.1.2	任务规范	239
12.1.3	任务时序图	239
12.1.4	最坏情况下的响应时间	241
12.1.5	任务实现	242
12.2	轮询架构	242
12.2.1	案例研究: 体温计	242
12.2.2	通用轮询架构	250

14.2.3 任务参数估计.....	299	16.6.2 具有更早截止时间的 任务.....	346
14.3 多任务资源共享.....	301	16.6.3 示例.....	347
14.3.1 资源死锁.....	301	16.6.4 优先级相同的任务.....	350
14.3.2 优先级反转.....	302	16.7 习题.....	351
14.4 解决资源死锁.....	304	第 17 章 实时调度：偶发服务器	
14.4.1 死锁预防.....	304	程序	355
14.4.2 死锁检测.....	304	17.1 偶发任务.....	355
14.4.3 死锁避免.....	305	17.2 偶发服务器程序.....	356
14.5 解决优先级反转.....	307	17.2.1 偶发服务器程序的相关 任务设计.....	356
14.5.1 优先级继承协议.....	307	17.2.2 验收测试.....	358
14.5.2 最高锁协议.....	309	17.3 朴素偶发服务器程序.....	360
14.5.3 优先级天花板协议.....	310	17.3.1 任务设计.....	361
14.6 习题.....	312	17.3.2 验收测试.....	361
第 15 章 实时调度：时钟驱动		17.4 固定优先级的偶发服务器 程序.....	362
方式	315	17.5 具有动态优先级的偶发 服务器程序.....	368
15.1 周期性调度简介.....	315	17.6 习题.....	371
15.1.1 若干假设.....	315	第 IV 部分 实现模式	
15.1.2 可抢占的非周期作业.....	316	第 18 章 资源共享	375
15.2 点对点时钟驱动调度.....	317	18.1 共享变量.....	375
15.2.1 点对点时钟驱动调度 程序.....	318	18.2 共享的内存.....	378
15.2.2 执行开销.....	319	18.2.1 映射文件对象.....	380
15.3 基于帧的调度.....	320	18.2.2 共享的内存对象.....	382
15.3.1 帧尺寸的约束.....	320	18.3 信号量.....	384
15.3.2 健壮的基于帧的调度.....	323	18.3.1 任务同步.....	385
15.3.3 基于帧的调度程序.....	324	18.3.2 流控制.....	387
15.4 调度非周期作业.....	325	18.3.3 资源保护.....	387
15.5 拆分任务.....	328	18.3.4 信号量相关的 POSIX 函数.....	389
15.6 习题.....	330	18.3.5 信号量示例.....	390
第 16 章 实时调度：速率单调		18.4 互斥体.....	397
方式	331	18.4.1 互斥体使用模式.....	398
16.1 优先级分配.....	331	18.4.2 互斥体相关的 POSIX 函数.....	400
16.2 RMA 原则.....	332		
16.3 速率单调分析.....	335		
16.4 完成时间测试.....	336		
16.5 周期变换.....	339		
16.6 通用的可调度性分析.....	342		
16.6.1 具有阻塞时间项的任务.....	342		

18.4.3	使用互斥体的例子	401	21.5.1	示例：在不同进程中处理信号	457
18.5	条件变量	404	21.5.2	示例：控制任务服务器	460
18.5.1	栅栏同步	405	21.6	来自 ISR 的自旋锁和中断事件	463
18.5.2	生产者-消费者模式	410	21.6.1	POSIX 自旋锁	463
18.5.3	读-写锁	414	21.6.2	QNX 事件结构	464
18.6	习题	418	21.6.3	QNX 应用程序的中断处理	465
第 19 章	任务间通信：消息队列	421	21.6.4	示例：来自 ISR 的中断事件	466
19.1	消息队列简介	421	21.7	QNX 脉冲	473
19.2	消息队列静态结构和动态转移	421	21.7.1	QNX 同步消息传递	473
19.3	消息队列使用模式	424	21.7.2	QNX 异步脉冲发生机制	476
19.3.1	单向通信	424	21.7.3	分层消息传递模式	478
19.3.2	确认-单向通信	425	21.7.4	消息接收者的优先级继承	478
19.3.3	双向通信	426	21.7.5	示例：简单的计时管 理器	479
19.3.4	客户端-服务器之间的通信	427	21.8	习题	488
19.4	与消息队列相关的 POSIX 函数	430	第 22 章	软件计时器管理	489
19.5	使用消息队列的例子	432	22.1	硬件计时器和软件计时器	489
19.6	习题	437	22.2	软件计时管理器	491
第 20 章	任务间通信：管道	439	22.2.1	链接到专用计时器 ISR	492
20.1	管道简介	439	22.2.2	使用 OS 计时器	492
20.2	管道的静态结构和动态状态转移	439	22.3	计时轮	493
20.3	管道使用模式	442	22.3.1	精度误差	495
20.4	与管道相关的 POSIX 函数	442	22.3.2	宽范围计时器	497
20.4.1	多个作者和读者	444	22.4	分层计时轮	497
20.4.2	在管道上 POSIX 的选择操作	445	22.4.1	计时轮参考上下文和计时器管理	498
20.5	使用管道的示例	446	22.4.2	实现	500
20.6	习题	451	22.5	习题	502
第 21 章	任务间通信：发送信号	453	第 23 章	QNX 资源管理	505
21.1	POSIX 信号简介	453	23.1	QNX 资源管理简介	505
21.2	信号处理	455	23.2	资源管理器体系架构	506
21.3	信号向量和处理程序	455	23.2.1	控制结构	507
21.4	POSIX 信号函数	456	23.2.2	关键数据结构	508
21.5	POSIX 信号的 QNX 实现	457			

23.3 示例 1: 作为资源管理器的 计算器.....	508
23.3.1 上层结构.....	509
23.3.2 处理来自客户端的消息.....	510
23.3.3 注册到进程管理器.....	517
23.3.4 使用资源管理器.....	518
23.4 示例 2: 设备驱动程序.....	520
23.4.1 操纵杆.....	520
23.4.2 LED 指示灯.....	521
23.4.3 基于轮询的输入事件 检测.....	523
23.5 习题.....	526
参考文献.....	527