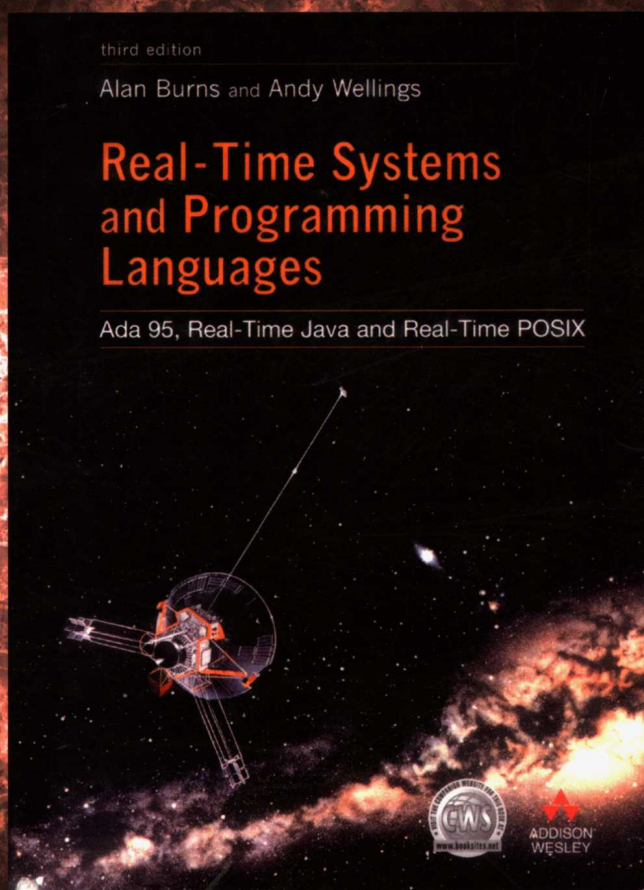


原书第3版

计 算 机 科 学 丛 书

实时系统与编程语言

(英) Alan Burns Andy Wellings 著 王振宇 陈利 等译



Real-Time Systems and Programming Languages

Ada 95, Real-Time Java and Real-Time POSIX, Third Edition



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

计

算

机

科

(

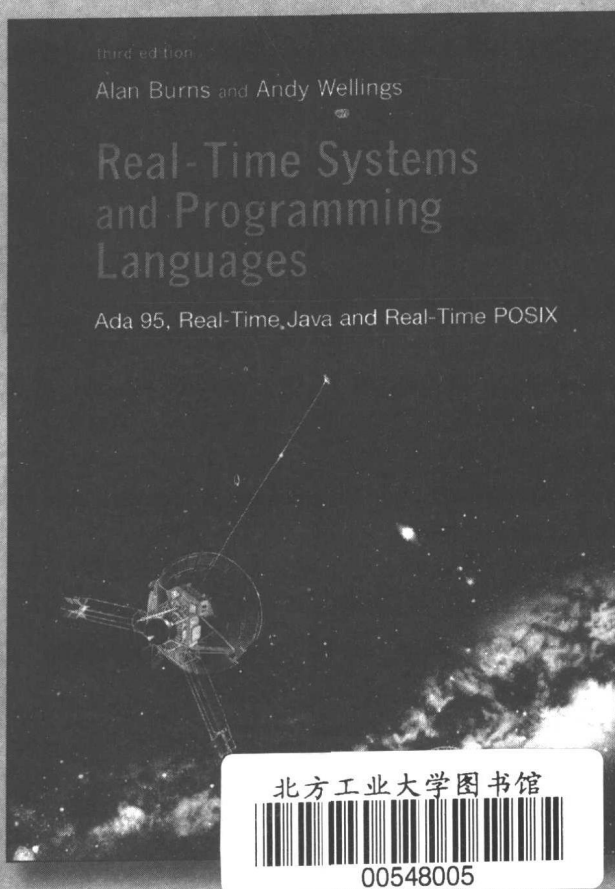
TP316.2
12

书

原书第3版

实时系统与编程语言

(英) Alan Burns Andy Wellings 著 王振宇 陈利 等译



Real-Time Systems and Programming Languages
Ada 95, Real-Time Java and Real-Time POSIX, Third Edition



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书全面论述实时系统、嵌入式系统和分布式系统的特征,深入分析设计和实现实时嵌入式系统的需求,并讨论了当前的编程语言和操作系统如何满足这些需求,重点介绍Ada 95、实时Java和实时POSIX。本书还覆盖了在实时领域的最新成果,包括实时CORBA。

本书在国外是实时系统方面的畅销教材,涵盖的内容广泛,适合作为高等院校计算机专业的教材,供高年级本科生和研究生使用。

Alan Burns and Andy Wellings: Real-Time Systems and Programming Languages: Ada 95, Real-Time Java and Real-Time POSIX (Third Edition) (ISBN: 0201729881)

Copyright © 1989, 2001 by Pearson Education Limited.

This translation of Real-Time Systems and Programming Languages: Ada 95, Real-Time Java and Real-Time POSIX (Third Edition) is published by arrangement with Pearson Education Limited.

本书中文简体字版由英国Pearson Education培生教育出版集团授权机械工业出版社和中信出版社出版。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2002-0605

图书在版编目(CIP)数据

实时系统与编程语言: 原书第3版 / (英)波恩斯(Burns, A.), (英)威林斯(Wellings, A.) 著; 王振宇等译. - 北京: 机械工业出版社, 2004.4

(计算机科学丛书)

书名原文: Real-Time Systems and Programming Languages: Ada 95, Real-Time Java and Real-Time POSIX (Third Edition)

ISBN 7-111-13987-9

I. 实… II. ①波… ②威… ③王… III. ①实时操作系统 ②程序语言 IV. ①TP316.2 ②TP312

中国版本图书馆CIP数据核字(2004)第013761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: 李 英

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2004年4月第1版第1次印刷

787mm × 1092mm 1/16 · 37.25印张

印数: 0 001 - 4 000册

定价: 59.00元

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，不仅擘划了研究的范畴，还揭橥了学术的源变，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，有的还专诚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的著名学者组成“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,我们的联系方法如下:

电子邮件: hzedu@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
周立柱
范明
袁崇义
谢希仁

王珊
吕建
李伟琴
陆丽娜
周克定
郑国梁
高传善
裘宗燕

冯博琴
孙玉芳
李师贤
陆鑫达
周傲英
施伯乐
梅宏
戴葵

史忠植
吴世忠
李建中
陈向群
孟小峰
钟玉琢
程旭

史美林
吴时霖
杨冬青
周伯生
岳丽华
唐世渭
程时端

秘 书 组

武卫东

温莉芳

刘江

杨海玲

译者序

Alan Burns 和Andy Wellings的这本《实时系统与编程语言》于1989年出第1版, 1997年出第2版, 2001年出第3版, 2003年则是第3版的第3次印刷, 可见它是一本受欢迎的书。Alan Burns 和Andy Wellings 都在英国约克 (York) 大学计算机科学系工作。Burns教授是IEEE实时系统委员会的主席 (2001-2003)。Andy Wellings 教授是*Software Practice and Experience*杂志的欧洲主编。

书中全面论述了实时系统、嵌入式系统和分布式系统的特征, 深入分析了设计和实现实时嵌入式系统的需求, 批判性地介绍了几种编程语言和操作系统对设计和实现实时系统的支持, 重点是Ada 95、实时Java和实时POSIX。书中覆盖了此领域的最新成果, 包括实时CORBA。

在这本书的翻译过程中, 关于若干术语的译法碰到不少问题。这一方面是由于国外一些作者和机构对这些术语的用法历来就有不同, 另一方面是由于中西文化背景的差异。我们的原则是这本书中的译法要尽可能统一, 不致使读者混淆, 并在必要时加注原文。另外, 做到“信、达、雅”全面兼顾, 当然很好, 但由于种种原因无法兼顾的时候, 只有牺牲“雅”了。相信翻译过科技著作的人会同意这种取舍。

原书作者针对第3版给出了勘误表, 见:

<http://www.booksites.net/burns>

译文已据此做了相应更改。遗憾的是还发现另外一些错误, 译文中也改正了。

参加本书翻译工作的除封面署名外还有: 王志海 (武汉理工大学)、张立 (武汉化工学院)、陈靖 (中船重工709研究所)、魏非 (中船重工709研究所)、余扬 (华南理工大学)、徐冠勇 (中船重工709研究所)、李扬 (中船重工709研究所)、李任 (中船重工709研究所) 等。

王振宇 (中船重工709研究所)

陈利 (华中师范大学)

2003年9月1日

前言

1981年，一个软件错误导致一个静止的机器人突然移动，并以极快的速度冲到操作区的边缘，附近的一个工人被撞死。这只是嵌入式实时系统灾难的一个例子。但是这并非一个孤立的意外事件。每个月，《Software Engineering Notes》（软件工程通讯）都有几页关于实时系统的故障将公众或环境推入危险境地的例子。这些认真描述的事件证明了需要对嵌入式系统有一个系统范围的观点。确实，有人主张说现在需要将实时系统作为一个单独的工程学科来认识。本书就是要为建立这个学科做出贡献。当然，它不可能覆盖有关实时系统工程研究的所有问题，然而，它确实提供了对此领域使用的编程语言和操作系统标准的全面描述和评估。特别强调的重点是语言原语及其在产生可靠、安全、可依赖软件中的作用。

本书面向的读者

本书针对计算机科学和相关专业高年级本科生和硕士研究生，也适合于专业软件工程师和实时系统工程师。本书假设读者具有像Pascal这种顺序编程语言的知识，并熟悉软件工程的基本原则。书中所用材料是作者过去多年里在各种大学和相关行业所开课程的内容，这些课程都是专门讲述实时系统和编程语言的。

结构和内容

为了各章的连贯性，本书详细地研究了4种编程语言：Ada、Java、occam2和C。之所以选中这些语言，是因为软件生产中真正使用这些语言。由于C是一个顺序语言，它要连同POSIX操作系统的接口一起使用（尤其是实时扩展）。本书也讨论其他的理论性或实验性语言，只要它们提供核心语言中没有的原语。这些语言的实践人员应当可找到他们需要的足够材料。作者相信，像对Ada或Java这样的语言的全面评价只能通过对它们进行比较性研究才能得出。

本书总共有18章，前13章分成五部分。第1章至第4章是一个深入的引言，提出了实时系统的特征和需求，然后给出此类系统的设计概述。设计不是本书的关注重点，然而它对于在适宜的环境中讨论实现问题是很重要的——第2章提供这样的背景。这一章还研究语言评估的一般标准。第3章和第4章通过关于小型编程和大型编程的讨论研究基本语言结构。这些章还介绍Ada、Java、occam2和C。熟悉这些语言和实时系统基本特性的读者可以快速地浏览这四章。对于其他读者，这些材料有助于使本书更加自成体系。

第5章和第6章讨论可靠软件部件的生产问题。虽然考虑到了故障预防，但重点主要集中在容错，研究了向前和向后两种出错恢复技术。第6章讨论异常处理，描述恢复和终止两种模型，以及在Ada、Java、CHILL、CLU、C++和Mesa中的语言原语。

实时系统的固有特征是并发，所以研究编程语言的这个方面是很重要的。第7章介绍进程的概念并概述语言和操作系统设计者使用的许多不同模型。在接下来的两章中研究进程之间的通信，首先描述共享变量方法，包括信号量、管程、互斥锁和保护对象的使用。在现代编程语言中，基于消息的模型也是很重要的，按它们进行通信和同步予以组合。第9章涵盖这些

模型，并特别关注Ada和occam2的会合原语。

对于在此书中是应当首先研究可靠性还是研究并发性的问题是有争论的。两位作者都试着反转次序，发现在这两种方法之间很难选择。事实上，这本书可以以任一种方式使用，只有一、两个主题是“不合适”的。首先讲解可靠性的决定反映了作者关于安全性是实时系统的根本需求的信念。

下一部分包括第10章和第11章。通常将系统进程之间的关系描述为合作关系（实现一个共同目标）或竞争关系（获得共享资源）。第10章通过描述如何将可靠的进程合作进行编程扩展了前面关于容错的讨论。这个讨论的核心是原子动作的概念和异步事件处理技术。接下来的一章研究竞争进程，评价了各种语言特征。这里的一个重要问题是并发模型里的条件同步和回避同步之间的区别。

时间性需求是实时系统的区别性特征。第12章和第13章详细讨论这些需求和用以满足需求的语言设施以及实现策略。硬实时系统的时间性约束是必须满足的；软实时系统可以偶尔不满足。二者都是在时限调度的背景下讨论的。优先级概念是同基于抢占式优先级系统的可调度性分析一起讨论的。

剩下的几章本质上是独立的。近期在硬件和通信技术方面的进展已经使分布式计算机系统在嵌入式应用领域成为单处理器和集中式系统的可行的替代品。虽然在某些方面，分布可被看作是实现问题；然而，当应用被分布时，出现的问题提出了超出实现细节的基础性问题。第14章研究相关的四个方面：语言支持、处理器和通信失效情况下的可靠性、分布式控制算法和分布式调度。把这些内容独立成章，目的是使上短期课程的学生可以跳过它。

许多实时系统的一个重要需求是加进来的外部设备必须作为应用软件的一部分编程（也就是说必须予以控制）。这种低级编程同作为软件工程特征的软件生产的抽象方法很不相同。第15章研究低级设施能够被成功地并入高级语言的方式。

对于实时系统的一个普遍错误概念是认为它们必须是高效的，这本身是不对的。实时系统必须满足时间约束（和可靠性需求）；高效的实现是扩展可能性范围的一种手段，但其本身不是目的。第16章概述了执行环境在获得高效可预测的实现中的作用。

本书最后的一章是用Ada实现的一个个案研究，使用了矿井控制系统的例子。一个经过裁减的个案研究不可能说明前面各章涵盖的所有问题，特别是没有涉及大小和复杂性等因素。然而，这个个案研究确实覆盖了实时系统的许多重要方面。

各章都有小结和相关阅读材料清单，多数章还有练习。选择这些练习是为了帮助读者巩固对每一章内容的理解。它们大体上代表了作者进行评估所使用过的练习。

Ada、Java、occam2和C

本书中Ada 95的例子符合ISO/ANSI标准。Java例子符合Java 2平台和Java扩展的实时规格说明（在本书中称为实时Java）。occam2例子符合INMOS给出的occam2定义。C例子符合ANSI C，POSIX原语是在POSIX.1、POSIX.1b、POSIX.1c、POSIX.1d和POSIX.13定义中给出的原语。

为方便四种语言的识别，使用了不同的表现风格。Ada使用加粗小写字母的关键字，程序标识符则以大小写混合方式给出。occam2的关键字用不加粗大写字母，标识符用混合大小写字母。C的关键字不加粗，标识符用小写字母。为将Java同C区分，Java关键字是加粗的，标

识符是混合大小写字母。

对第2版的修改

在第2版中，我们从Ada 83移到Ada 95，从Modula-2移到了C和POSIX。第3版根据实时领域的下列发展进行了改进：

- Java通过实时Java扩展成为一个实时语言。因此本书将Java作为核心语言之一对待。
- POSIX标准中补充的新实时功能，尤其是执行时间监控（Execution Time Monitoring）和偶发服务器（Sporadic Server）。第12章和第13章的更新反映了标准的这些修改。
- 为解决实时问题对COBRA的建议扩展。CORBA显然有理由成为一个主要议题，但详细讨论它超出了本书的范围。然而，第14章的更新还是反映了CORBA方法。

我们已经接受了使用本书教学的读者关于包含更多调度方面材料的建议。

教学辅助材料

从下列网站可得到本教科书的更多支持材料：<http://www.booksites.net/burns>。本书许多部分有投影胶片可用，还提供许多练习的答案。我们将随时添加更多的练习、适当的新例子和补充教学材料。使用本书的教师/讲师帮助完成了这些网页。

关于约克大学的实时系统研究

Alan Burns和Andy Wellings是英国约克大学计算机系实时系统研究组的成员。这个小组对实时系统的设计、实现和分析的所有方面进行研究。这个小组尤其致力于：形式化和结构化开发方法、调度理论、重用、语言设计、内核设计、通信协议、分布和并行体系结构、程序代码分析。这个研究组的宗旨是进行基础性研究，并将现代技术、方法和工具带入工程实践。研究组的应用领域包括空间和航空电气系统、引擎控制器、汽车控制和多媒体系统。可通过<http://www.cs.york.ac.uk/rts/>找到这个小组活动的进一步信息。

第1版的致谢

本书中的材料是在过去五年里建立起来的，并在英国布拉德福德大学（Bradford）和约克大学计算机科学和电子学系向三年级本科生和研究生讲授过。我们要感谢他们对本书最终出版的贡献，没有他们将不可能写成这本书。

许多人读过本书的第一稿并提出意见。我们特别要感谢Martin Atkins、Chris Hoggarth、Andy Hutcheon、Andrew Lister和Jim Welsh。我们还要感谢我们在大学里的同事，他们为我们提供了激励性的环境和许多有价值的讨论，特别是Ljerka Beus-Dukic、Geoff Davies、John McDermid、Gary Morgan、Rick Pack、Rob Stone和Hussein Zedan。

1988年Alan Burns在澳大利亚昆士兰（Queensland）大学和美国休斯顿（Houston）大学进行假期研修。我们感谢在这里的所有同事，特别是Andrew Lister、Charles McKay和Pat Rogers。

如果不用通过JANET传送的电子邮件，也不可能写成这本书。我们要感谢英国大学大理石会和科学工程研究基金会的计算机处，它们提供了这种无价的服务。

最后，我们还要特别感谢Sylvia Holmes和Carol Burns，感谢Sylvia在最终手稿上进行辛苦的校对，感谢Carol容许我们进行许多晚间的会议和讨论。

第2版的致谢

许多人帮助我们出版本书第2版。我们要特别感谢Alejandro Alonso、Angel Alvarez、Sergio Arevalo、Neil Audsley、Martin Dorey、Michael Gonzalez Harbour、Stuart Mitchell、Gary Morgan、Offer Pazy和Juan de la Puente。

第3版的致谢

我们要感谢实时Java专家组，他们在建立实时Java规格说明时采用了开放的方式。还感谢Angel Alvarez、Jose Alvarez、Neil Audsley、Iain Bate、Jorge Diaz-Herrera、David Duke、Alan Grigg、Ian Hayes、George Lima、Greg Murphy、Peter Puschner 和Pat Rogers，在写这一版时，他们向我们提供了各种形式的帮助。

我们还想感谢技术评审员Jorge Diaz-Herrera、Jörgen Hansson和Robert Holton，他们对本版的初稿提出了很有价值的意见。

最后，我们要感谢所有向本书第2版提出意见的人。

Alan Burns和Andy Wellings

于英国约克大学

2000年11月

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第1章 实时系统引论	1
1.1 实时系统的定义	1
1.2 实时系统的例子	2
1.2.1 过程控制	2
1.2.2 制造业	3
1.2.3 通信、指挥与控制	4
1.2.4 广义嵌入式计算机系统	5
1.3 实时系统的特征	5
1.3.1 大且复杂	5
1.3.2 实数处理	6
1.3.3 极其可靠和安全	7
1.3.4 独立系统部件的并发控制	7
1.3.5 实时设施	8
1.3.6 同硬件接口的交互	8
1.3.7 高效的实现和执行环境	9
小结	9
相关阅读材料	9
第2章 设计实时系统	11
2.1 记号系统的级别	12
2.2 需求规格说明	12
2.3 设计活动	13
2.3.1 封装	13
2.3.2 内聚和耦合	14
2.3.3 形式化方法	14
2.4 设计方法	15
2.4.1 JSD	16
2.4.2 Mascot3	17
2.4.3 HRT-HOOD	18
2.4.4 统一建模语言 (UML)	19
2.5 实现	19
2.5.1 汇编语言	20
2.5.2 顺序系统实现语言	20
2.5.3 高级并发编程语言	20
2.5.4 通用语言设计标准	21
2.6 测试	22
2.7 原型建造	23
2.8 人机交互	24
2.9 设计的管理	25
小结	26
相关阅读材料	27
练习	27
第3章 小型编程	29
3.1 Ada、Java、C和occam2概述	29
3.2 词法约定	29
3.3 整体风格	30
3.4 数据类型	31
3.4.1 离散类型	32
3.4.2 实数	33
3.4.3 结构化数据类型	35
3.4.4 动态数据类型和指针	37
3.4.5 文件	39
3.5 控制结构	39
3.5.1 顺序结构	39
3.5.2 判断结构	40
3.5.3 循环结构	43
3.6 子程序	46
3.6.1 参数传递模式和机制	46
3.6.2 过程	47
3.6.3 函数	49
3.6.4 子程序指针	50
3.6.5 插入式展开	51
小结	51
相关阅读材料	52
练习	52

第4章 大型编程	55	6.1.1 反常返回值	101
4.1 信息隐藏	55	6.1.2 强迫性分支	102
4.2 分别编译	59	6.1.3 非局部go to	102
4.3 抽象数据类型	60	6.1.4 过程变量	103
4.4 面向对象编程	61	6.2 现代异常处理	104
4.4.1 OOP和Ada	62	6.2.1 异常及其表示	104
4.4.2 OOP和Java	65	6.2.2 异常处理程序的定义域	105
4.4.3 继承和Java	66	6.2.3 异常传播	107
4.4.4 对象类	69	6.2.4 恢复模型与终止模型的对比	107
4.5 可重用性	70	6.3 Ada、Java和C中的异常处理	110
4.5.1 Ada类属编程	70	6.3.1 Ada	110
4.5.2 Java中的接口	73	6.3.2 Java	117
小结	75	6.3.3 C	123
相关阅读材料	76	6.4 其他语言中的异常处理	124
练习	76	6.4.1 CHILL	124
第5章 可靠性和容错	77	6.4.2 CLU	125
5.1 可靠性、失效和故障	78	6.4.3 Mesa	126
5.2 失效模式	79	6.4.4 C++	126
5.3 故障预防与容错	80	6.5 恢复块和异常	127
5.3.1 故障预防	80	小结	129
5.3.2 容错	81	相关阅读材料	130
5.3.3 冗余	82	练习	130
5.4 N版本程序设计	82	第7章 并发编程	135
5.4.1 表决比较	84	7.1 进程概念	135
5.4.2 N版本程序设计的主要问题	85	7.2 并发执行	138
5.5 软件动态冗余	86	7.3 进程表示	140
5.5.1 出错检测	86	7.3.1 合作例程	140
5.5.2 损害隔离和评估	87	7.3.2 分叉与汇合	140
5.5.3 出错恢复	88	7.3.3 cobegin	142
5.5.4 故障处理和继续服务	90	7.3.4 显式进程声明	142
5.6 软件容错的恢复块方法	90	7.3.5 occam2的并发执行	143
5.7 N版本程序设计和恢复块的比较	92	7.3.6 Ada的并发执行	144
5.8 动态冗余和异常	93	7.3.7 Java的并发执行	148
5.9 软件可靠性的测量和预测	95	7.3.8 Ada、Java和occam2的比较	154
5.10 安全性、可靠性和可依赖性	95	7.3.9 POSIX的并发执行	155
小结	97	7.4 一个简单的嵌入式系统	159
相关阅读材料	98	小结	164
练习	98	相关阅读材料	165
第6章 异常和异常处理	101	练习	165
6.1 老式实时语言中的异常处理	101	第8章 基于共享变量的同步和通信	169

8.1 互斥和条件同步	169
8.2 忙等待	170
8.3 挂起和恢复	175
8.4 信号量	177
8.4.1 挂起进程	179
8.4.2 实现	180
8.4.3 活性	181
8.4.4 二元信号量和定量信号量	182
8.4.5 Ada信号量编程举例	182
8.4.6 使用C和POSIX的信号量编程	184
8.4.7 对信号量的批评	186
8.5 条件临界区	186
8.6 管程	187
8.6.1 Modula-1	189
8.6.2 Mesa	190
8.6.3 POSIX互斥锁和条件变量	192
8.6.4 嵌套管程调用	194
8.6.5 对管程的批评	194
8.7 保护对象	195
8.7.1 入口调用和屏障	197
8.7.2 保护对象和标记类型	199
8.8 同步方法	200
8.8.1 等待和通知	201
8.8.2 继承和同步	206
小结	208
相关阅读材料	210
练习	210
第9章 基于消息的同步与通信	219
9.1 进程同步	219
9.2 进程指名和消息结构	220
9.3 Ada和occam2的消息传递语义	221
9.3.1 occam2模型	221
9.3.2 Ada模型	222
9.3.3 异常处理和会合	225
9.4 选择性等待	226
9.4.1 occam2的ALT	226
9.4.2 Ada的Select语句	230
9.4.3 不确定性、选择性等待和同步 原语	232

9.5 POSIX消息	233
9.6 CHILL语言	236
9.7 远程过程调用	238
小结	239
相关阅读材料	240
练习	240
第10章 原子动作、并发进程和可靠性	247
10.1 原子动作	247
10.1.1 两阶段原子动作	248
10.1.2 原子事务	249
10.1.3 对原子动作的需求	249
10.2 并发语言中的原子动作	250
10.2.1 信号量	251
10.2.2 管程	252
10.2.3 用Ada实现原子动作	253
10.2.4 用Java实现原子动作	254
10.2.5 用occam2实现原子动作	258
10.2.6 原子动作的语言框架	259
10.3 原子动作和向后出错恢复	260
10.3.1 会话	260
10.3.2 对话和会谈	261
10.4 原子动作和向前出错恢复	262
10.4.1 并发引发的异常的分辨	263
10.4.2 异常和内部原子动作	263
10.5 异步通知	264
10.6 POSIX信号	265
10.6.1 阻塞信号	267
10.6.2 处理信号	267
10.6.3 忽略信号	268
10.6.4 生成信号	268
10.6.5 一个POSIX信号的简单例子	268
10.6.6 信号和线程	269
10.6.7 POSIX和原子动作	270
10.7 实时Java中的异步事件处理	271
10.8 Ada中的异步控制转移	272
10.8.1 异常和ATC	274
10.8.2 Ada和原子动作	274
10.9 实时Java中的异步控制转移	281
小结	291

相关阅读材料	292	12.7 时序作用域的语言支持	342
练习	292	12.7.1 Ada, occam2和C/POSIX	343
第11章 资源控制	297	12.7.2 实时Euclid和Pearl	344
11.1 资源控制和原子动作	297	12.7.3 实时Java	346
11.2 资源管理	298	12.7.4 DPS	349
11.3 表达能力和易用性	298	12.7.5 Esterel	350
11.3.1 请求类型	299	12.8 容错	351
11.3.2 请求顺序	301	12.8.1 时间性错误检测和向前出错恢复	352
11.3.3 服务器状态	301	12.8.2 时间性错误检测和向后出错恢复	358
11.3.4 请求参数	301	12.8.3 模式改变和基于事件的重配置	359
11.3.5 请求者优先级	305	小结	361
11.3.6 小结	306	相关阅读材料	362
11.4 重排队设施	306	练习	363
11.4.1 重排队的语义	309	第13章 调度	365
11.4.2 重排队到其他入口	310	13.1 简单进程模型	365
11.5 不对称指名和安全性	312	13.2 循环执行方法	366
11.6 资源的使用	313	13.3 基于进程的调度	368
11.7 死锁	313	13.3.1 调度方法	368
11.7.1 死锁发生的必要条件	313	13.3.2 抢占和非抢占	368
11.7.2 处理死锁的方法	314	13.3.3 FPS和速率单调优先级分配	368
小结	316	13.4 基于利用率的可调度性测试	369
相关阅读材料	317	13.5 FPS的响应时间分析	372
练习	317	13.6 EDF的响应时间分析	375
第12章 实时设施	321	13.7 最坏情况执行时间	376
12.1 时间的概念	321	13.8 偶发和非周期进程	377
12.2 时钟访问	323	13.8.1 硬进程和软进程	377
12.2.1 occam2中的TIMER	323	13.8.2 非周期进程和固定优先级服务器	378
12.2.2 Ada的时钟包	324	13.8.3 非周期进程和EDF服务器	378
12.2.3 实时Java中的时钟	326	13.9 $D < T$ 的进程系统	379
12.2.4 C和POSIX中的时钟	329	13.10 进程交互和阻塞	380
12.3 进程延迟	330	13.11 高限优先级协议	383
12.3.1 相对延迟	330	13.11.1 立即高限优先级协议	385
12.3.2 绝对延迟	330	13.11.2 高限协议、互斥和死锁	385
12.4 超时的编程	332	13.11.3 阻塞和EDF	386
12.4.1 共享变量通信和超时	332	13.12 一个可扩充的进程模型	386
12.4.2 消息传递和超时	333	13.12.1 合作调度	386
12.4.3 动作上的超时	337	13.12.2 启动抖动	387
12.5 规定时间性需求	339	13.12.3 任意的时限	389
12.6 时序作用域	340	13.12.4 容错	390

13.12.5 引入偏移量	390	练习	444
13.12.6 优先级分配	392	第15章 低级编程	447
13.13 动态系统和联机分析	392	15.1 硬件输入/输出机制	447
13.14 基于优先级系统的编程	393	15.1.1 状态驱动	448
13.14.1 Ada	394	15.1.2 中断驱动	448
13.14.2 POSIX	397	15.1.3 中断驱动设备所需的要素	449
13.14.3 实时Java	399	15.1.4 一个简单的I/O系统的例子	451
13.14.4 实时Java的其他设施	402	15.2 语言要求	452
小结	402	15.2.1 模块性和封装设施	452
相关阅读材料	403	15.2.2 设备处理的抽象模型	453
练习	404	15.3 Modula-1	454
第14章 分布式系统	409	15.3.1 设备寄存器的寻址和操纵	454
14.1 分布式系统的定义	409	15.3.2 中断处理	455
14.2 论题一览	411	15.3.3 一个终端驱动程序例子	456
14.3 语言支持	412	15.3.4 Modula-1设备驱动方法的问题	459
14.3.1 远程过程调用	412	15.4 Ada	460
14.3.2 分布式对象模型	413	15.4.1 设备寄存器的寻址和操作	460
14.4 分布式编程系统和环境	413	15.4.2 中断处理	462
14.4.1 occam2	414	15.4.3 一个简单的驱动程序例子	464
14.4.2 Ada	416	15.4.4 通过特别指令访问I/O设备	467
14.4.3 Java	419	15.5 实时Java	468
14.4.4 CORBA	421	15.5.1 设备寄存器的寻址和操纵	468
14.5 可靠性	423	15.5.2 中断处理	470
14.5.1 开放系统互连	424	15.6 occam2	470
14.5.2 TCP/IP层	426	15.6.1 一个设备驱动程序例子	472
14.5.3 轻量级协议和局域网	426	15.6.2 occam2设备驱动的困难	476
14.5.4 组通信协议	427	15.7 C和老式实时语言	476
14.5.5 处理器失效	428	15.8 设备驱动程序的调度	477
14.6 分布式算法	432	15.9 存储管理	479
14.6.1 分布式环境中的事件排序	432	15.9.1 堆管理	479
14.6.2 全局时间的实现	433	15.9.2 栈管理	484
14.6.3 实现稳定存储	434	小结	484
14.6.4 故障性进程出现时达成一致	435	相关阅读材料	485
14.7 分布式环境中的时限调度	437	练习	485
14.7.1 分配	437	第16章 执行环境	491
14.7.2 调度对通信链路的访问	439	16.1 执行环境的作用	491
14.7.3 整体调度	441	16.2 剪裁执行环境	493
小结	442	16.2.1 Ada中的受限任务	493
相关阅读材料	444	16.2.2 POSIX	495

16.3 调度模型	495	17.3.4 数据记录器和操作员控制台	509
16.3.1 非微小的上下文切换时间的建模	496	17.4 物理体系结构设计	509
16.3.2 偶发进程的建模	497	17.5 翻译到Ada	511
16.3.3 实时时钟处理程序的建模	497	17.5.1 水泵控制器对象	512
16.3.4 高速缓存对最坏情况执行时间分 析的影响	499	17.5.2 环境监控	520
16.4 硬件支持	499	17.5.3 气流传感器处理对象	523
16.4.1 传输机和occam2	500	17.5.4 CO传感器处理对象	524
16.4.2 ATAC和Ada	500	17.5.5 数据记录器	525
小结	501	17.5.6 操作员控制台	526
相关阅读材料	501	17.6 容错和分布	526
练习	502	17.6.1 设计错误	526
第17章 Ada案例研究	503	17.6.2 处理器和通信失效	527
17.1 矿井排水	503	17.6.3 其他硬件失效	527
17.1.1 功能需求	503	小结	528
17.1.2 非功能需求	504	相关阅读材料	528
17.2 HRT-HOOD设计方法	506	练习	528
17.3 逻辑体系结构设计	506	第18章 结论	529
17.3.1 第一级分解	507	附录 实时Java规格说明	533
17.3.2 水泵控制器	507	参考文献	553
17.3.3 环境监控器	509	索引	565