

<<Java 实时编程>>

图书基本信息

书名：<<Java 实时编程>>

13位ISBN编号：9787111292821

10位ISBN编号：7111292820

出版时间：2010年4月

出版时间：机械工业出版社

作者：Eric J. Bruno, Greg Bollella

页数：270

译者：田思源

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Java 实时编程>>

前言

将广泛使用的Java应用于实时领域，是一项有益的探索。

Java因其自身特点，如垃圾回收和即时编译，并不适合实时应用。

从另一个角度看，Java广大的用户群和成熟的开发环境，对于降低实时系统采用专用语言和操作系统环境所造成的开发门槛，有着相当的吸引力。

Java实时规范（RTSJ，也称为JSR-001）和Java实时系统（Java RTS）是在这个方向的一个尝试，并取得了不错的进展。

目前，Java RTS还有其局限，例如，不支持硬实时。

本书介绍了实时系统的基本原理和概念、Java的垃圾回收、如何使用Java RTS的API 解决实际问题、实时垃圾回收、线程调度和分派、新的RTSJ内存模型，以及异步事件处理和异步控制转移。

本书第三部分给出了一个用Java RTS实现的证券交易系统的例子，以及从非实时Java到RTS的转变过程和它们之间的差别。

同时，还介绍了开发用到的有关工具，主要包括基于Netbeans和Eclipse的集成开发环境、线程调度展示台和Java RTS Dtrace 探针。

本书基于RTSJ 1.0.2和Java RTS 2.1。

Java RTS 2.2已于2009年9月发布。

希望能看到Java RTS的进一步发展和真正的实际应用。

在翻译过程中，得到了机械工业出版社华章公司的朋友陈冀康的大力帮助，在此表示感谢。

最后，我要感谢外甥女妞妞对我的支持和帮助。

由于时间有限，书中难免存在译释不当之处，恳请读者批评指正。

<<Java 实时编程>>

内容概要

本书主要讲述使用Java RTS进行Java实时编程的各类主题。

主要内容包括3个部分：第一部分“实时计算概念”，明确定义计算领域中的实时概念，并讨论了许多实时应用程序设计和开发的理论；第二部分“高级Java RTS”，提供充足的示例代码来说明Java RTS的内部运作机制及其使用；第三部分“使用Java RTS”，讨论了实际的案例研究。

本书适合Java程序开发人员、实时软件开发人员，以及使用Java RTS编程的人员使用，也可供高等院校老师和学生参考。

<<Java 实时编程>>

作者简介

Eric J. Bruno, Sun公司系统工程师, 专门从事于金融界的Java RTS。
他是《Dr. Dobbs' s Jour-hal》的撰稿编辑, 并撰写其在线Java博客。
加入Sun之前, Eric曾在路透社工作, 在那里他用Java和C++开发实时交易系统、订单输入和路由系统, 以及实时新闻和报价数据源系统。

书籍目录

译者序 前言 致谢 作者简介 第一部分 实时计算概念 第1章 实时系统概述 1.1 实时系统质量 1.2 可预见性和确定性 1.3 实时计算 1.4 实时调度 1.5 参考资料 第2章 实时与Java SE 2.1 Java是实时语言吗 2.2 垃圾回收 2.3 Sun Java SE 6 HotSpot中的垃圾回收 2.4 实时垃圾回收算法 2.5 Java的困境 第3章 Java实时规范 3.1 实时Java 基础 3.2 RTSJ内部机制 3.3 可选的RTSJ工具 3.4 实时Java 的未来 第4章 Sun Java实时系统 4.1 Java RTS程序设计的前奏 4.2 安装Java RTS 4.3 确定性指南 4.4 命令行选项 第二部分 高级Java RTS 第5章 线程、调度与新内存模型 5.1 可调度对象 5.2 实时线程 5.3 内存模型 第6章 同步 6.1 资源共享 6.2 非等待线程通信 第7章 实时时钟API 7.1 Clock API 7.2 Java RTS高精度时间操作 7.3 叠加的股票数据提供的例子 第8章 异步事件 8.1 异步事件处理 8.2 创建一个异步事件处理程序 8.3 基于时间的事件 第9章 异步控制转移和线程终止 9.1 异步控制转移 9.2 异步线程终止 第10章 实时垃圾回收器内部机制 10.1 RTGC操作理论 10.2 RTGC内存分配 10.3 RTGC策略 第三部分 使用Java RTS 第11章 证券交易系统 11.1 股票市场 11.2 交易系统实现 11.3 Java SE版本 11.4 Java RTS 版本 11.5 Java RTS 没有堆的版本 11.6 应用程序配置 第12章 Java RTS工具 12.1 Java RTS开发 12.2 线程调度展示台 (TSV) 12.3 Java RTS DTrace探针 参考文献

章节摘录

1.3 实时计算 现在,你已经了解了实时是什么以及它意味着什么,现在要对它加以扩展。实时计算是针对真实世界的时间关键制约因素建设应用系统的研究与实践。实时系统必须在某个时间或针对某个最后期限响应外部(往往是物理的)真实世界的事件。实时系统整体通常包括硬件和软件。

传统上,实时系统是专门设计的系统,为特定使用而实施。

最近实时社区才专注于用通用计算机系统(硬件或软件)来解决实时问题。

今天,需要特定的专用硬件的实时系统已大部分消失了。

例如,现代的芯片组包括可用于实时应用的具有足够小的延迟精度的可编程中断控制器。

因此,对实时要求的支持已经转移到软件,即专门的调度程序和资源控制程序。

曾经蚀刻到特殊电路的算法现在用通用计算机的软件实现。

这并不是说,在实时系统中,不需要硬件支持。

例如,许多实时系统可能会要求访问可编程中断控制器的低延迟中断和调度、高精度时钟的精确计时、直接物理内存访问或一个高速缓存。

大多数现代的计算机硬件,包括服务器、工作站,甚至是台式机和笔记本电脑,支持这些要求。

蔗线是运行在这些硬件上的操作系统软件是否支持访问这些硬件设施。

事实上,操作系统可能直接通过其调度实现支持实时任务,或可能至少允许替代调度算法到位。

然而,许多通用操作系统的任务调度实现不同于实时系统的目标。

其他一些因素,如整个系统的吞吐量,前台应用程序的性能和GUI刷新率,可能是有利于个别任务的延迟要求。

事实上,在一个通用系统中,可能没有办法精确确定或衡量一个应用程序的延迟要求和实际结果。

然而,在通用的操作系统上,仍有可能实现实时行为,并满足实时任务的最后期限。

事实上,这是Java RTS和RTSJ宪章要解决的一个目标:在通用硬件和实时操作系统上的Java的实时行为。

在现实中,只有一些通用系统可以支持。

本章的余下部分概述了实时系统调度任务所涉及理论和机制。

必须明确的是,实时调度理论需要大量的数学用于描述以透彻理解。

这里有充分的理由:一个系统要求满足每一个行动的最后期限,否则,如果错过了,可能产生可怕的后果,你需要保证最大的精度。

表征和保证系统的行为,数学是做到这一点的唯一办法。

但是,我们会试图讨论这个问题,而不使你受限于过深的数学概念。

取而代之的是,使用类比、说明和图形表示使概念具体化,到令平均水平的程序员应该感觉很舒服的程度。

当然,也为那些对更深的数学和科学的问题感兴趣的人提供了进一步的阅读材料。

1.3.1 高速公路类比 一种最简单的描述实时系统的动态调度任务的方法是用高速公路作类比。

在开车时,我们都经历过大量交通流量的影响;即不可预测的大量时间花在了等待上,而不是向目标前进。

这种情况与实时系统或任何系统的任务调度有惊人的相似之处。

对于汽车交通,调度的项目是车辆,而它们共享的资源是道路空间。

<<Java 实时编程>>

编辑推荐

面向开发者和架构师的Java RTS权威指南。

面向转向实时系统领域的Java开发者和架构师，以及转向Java的实时系统开发者。

从头至尾逐步分析应用示例，确定其限制并讨论解决它们的API和设计模式。

作者为实时Java标准过程的前任组长和华尔街实时系统的顶级开发者。

Sun公司的Java实时系统（Java RTS）在多个广泛的环境中得到了实践验证，其中包括金融、控制系统、制造和防务。

Java RTS和RTSJ标准（JSR-001）消除了对于复杂、专用的实时语言和操作系统的需要，利用Java超常的生产率和为人所熟知的特点，节省了资金。

在实时Java编程中，Sun公司的两位实时编程专家介绍了开发者成功使用Java RTS及其API所需要的深入知识和实际代码示例，并阐明了用任何RTSJ兼容的环境进行实时编程的基础。

《Java 实时编程》的内容包括：实时原理和概念，以及实时应用设计和开发的独特要求。

如何把Java应用于实时环境。

垃圾回收和Java SE回收器。

使用Java RTS API尽可能高效地解决实际实时系统问题。

当前领先的Java RTS开发和调试工具。

实时垃圾回收、线程、调度和分派。

新的RTSJ内存模型。

异步事件处理和异步控制转移。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>