

分布式实时系统

张凤登 著



科学出版社

014032858

TP316.4

15

内容简介

随着信息科学技术的飞速发展,计算机系统的实时性要求越来越高,实时系统已成为许多领域(如工业控制、军事指挥、航空订票、金融交易等)中不可或缺的重要组成部分。本书系统地介绍了实时系统的概念、组成、分类、设计、实现、测试、维护等方面的知识,可作为高等院校计算机专业及相关专业的教材,也可供从事实时系统工作的工程技术人员参考。

本书共分8章。第1章介绍实时系统的概念、组成、分类、应用及实时系统的性能指标;第2章介绍实时系统的组成;第3章介绍实时系统的设计;第4章介绍实时系统的实现;第5章介绍实时系统的测试;第6章介绍实时系统的维护;第7章介绍实时系统的性能分析;第8章介绍实时系统的性能优化。

分布式实时系统

张凤登 著



科学出版社

北京



北航

C1721023

TP316.4
15

328580110

内 容 简 介

实时系统是在嵌入式系统、工业自动化系统和多媒体系统高度发展的基础上形成的一个新概念。本书采用认知领域的一些最新见解,以连贯、简洁、可理解的方式,系统地介绍了实时系统的产生背景、理论与技术基础,描述了分布式实时系统在架构层面的设计原理,并重点探讨了在预期负载和故障情况下强实时系统的设计、实现和评估方法。全书共分为10章,每章配有习题。

本书在编写过程中广泛吸取了实时系统设计方面的最新成果,全书内容自成体系,结构紧凑,前后呼应,具有一定的先进性、系统性和实用性。

本书可作为高等院校自动化、测控技术、信息工程、微电子、计算机、电气工程和机电一体化等专业高年级本科生、研究生的教材,也可作为从事嵌入式实时系统设计和应用的工程技术人员的参考书。

图书在版编目(CIP)数据

分布式实时系统/张凤登著. —北京:科学出版社,2014.3
ISBN 978-7-03-039313-5

I. ①分… II. ①张… III. ①分布式操作系统 IV. ①TP316.4

中国版本图书馆CIP数据核字(2013)第299809号

策划编辑:王 哲/责任编辑:王 哲 邢宝钦/责任校对:钟 洋
责任印制:张 倩/封面设计:迷底书装

科学出版社出版

北京东黄城根北街16号
邮政编码:100717

<http://www.sciencep.com>

北京市文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2014年3月第 一 版 开本:720×1 000 1/16

2014年3月第一次印刷 印张:21 1/2

字数:438 000

定价:96.00元

(如有印装质量问题,我社负责调换)

前 言

随着嵌入式系统、工业自动化系统和多媒体系统的不断发展,设备之间、人与系统之间的信息交换逐步实现网络化,各种网络化物理系统具备了自主获取和实时处理信息的能力,已被广泛应用于电力、冶金、化工、机械加工、食品加工和消费电子等领域,与人们的日常生产和生活息息相关。正因为如此,科技界已开始探索它们在安全性和可靠性要求极高的领域中的应用,如汽车、飞机、核电、交通、武器装备和航海航空等。学术界和工业界通过共同努力,逐步理解了很多应用实例的包含关系,并形成了一个新的科学概念——实时系统。这个概念与物理时间密切相关,不仅强调系统的控制运算、总线通信、错误诊断和结果预测能力,而且能够反映系统在预期负载和故障情况下的响应及时性和容错能力等,现已成为学术团体和工业组织的研究主题之一,也是当今网络化系统市场重要的技术促进因素。

为了确保实时系统安全可靠的运行,系统设计人员需要考虑很多因素,有时甚至颠覆了许多过去认为行之有效的系统设计原理。本书根据大量的工业实例,采用认知领域的一些最新见解,以连贯、简洁、可理解的方式,详细阐述了实时系统与时间之间的内在联系,从架构层面描述了这种系统的设计原理。

实时系统分为强实时系统和弱实时系统,本书侧重于讲述强实时系统方面的基本概念。全书共分为10章,第1、2章作为入门,主要介绍实时系统的基本概念,如实时环境、系统模型化;第3章着重描述Kopetz和Lampport在全局时钟同步和容错时钟同步方面的贡献;第4章给出实时系统的时间行为,如时间准确性、持久性、幂等性、复制确定性等;第5~8章利用故障、错误、失效和异常等概念,研究安全关键性系统的容错单元构造方法,探讨实时通信、实时操作系统、触发架构、CPU资源配置和实时调度算法等方面的重要见解;第9、10章讨论可依赖性实时系统的一般性设计和实现技术,并描述几种系统评估方法。围绕所讲内容,本书给出一些有代表性的设计实例。

本书的编写得到了资深学者、同事和科学出版社的大力支持。应启夏、张仁杰、缪学勤、吴勤勤为本书的组织结构和新术语的定义提出了很多宝贵意见;孟庆栋、周文杰、张勇、王闯、石秋蝉、廖振俭、华俊、尚雯雯、范科发、侯斌、王臻、张玮、胡羽、陈蕊、张晓霞、张大庆、李红雨等仔细阅读了部分或全部书稿,并提出了许多宝

贵的改进建议；科学出版社的王哲编辑在本书的体例格式和易读性方面给予了许多帮助。在此谨向他们致以衷心的感谢。

由于作者水平有限，书中难免存在不足之处，敬请读者批评指正。

前言

作者
2013年8月

本人，向之奇数，现代工业的迅速发展，特别是计算机工业的迅速发展，为人们的生活带来了巨大的便利。网络技术的发展，使得人们可以随时随地获取信息，进行交流和协作。工业控制系统的实时性要求越来越高，这对控制系统的实时性提出了更高的要求。

本书主要介绍了实时控制系统的概念、分类、组成、实时调度算法、实时数据库、实时通信以及实时控制系统的开发。本书力求做到概念清晰、重点突出、循序渐进、由浅入深、由理论到实践、由简单到复杂、由基础到提高、力求做到深入浅出、通俗易懂。本书可作为高等院校自动化专业及相关专业的教材，也可供从事实时控制系统的工程技术人员参考。

本书共分10章。第1章为绪论，介绍实时控制系统的概念、分类、组成以及实时控制系统的组成。第2章介绍实时调度算法，包括先来先服务调度算法、短作业优先调度算法、时间片轮转调度算法、实时调度算法。第3章介绍实时数据库，包括实时数据库的概念、组成、实时数据库的组成、实时数据库的组成。第4章介绍实时通信，包括实时通信的概念、组成、实时通信的组成、实时通信的组成。第5章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第6章介绍实时控制系统的开发，包括实时控制系统的开发、实时控制系统的开发、实时控制系统的开发。第7章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第8章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第9章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第10章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。

本书共分10章。第1章为绪论，介绍实时控制系统的概念、分类、组成以及实时控制系统的组成。第2章介绍实时调度算法，包括先来先服务调度算法、短作业优先调度算法、时间片轮转调度算法、实时调度算法。第3章介绍实时数据库，包括实时数据库的概念、组成、实时数据库的组成、实时数据库的组成。第4章介绍实时通信，包括实时通信的概念、组成、实时通信的组成、实时通信的组成。第5章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第6章介绍实时控制系统的开发，包括实时控制系统的开发、实时控制系统的开发、实时控制系统的开发。第7章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第8章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第9章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。第10章介绍实时控制系统的组成，包括实时控制系统的组成、实时控制系统的组成、实时控制系统的组成。

本书可作为高等院校自动化专业及相关专业的教材，也可供从事实时控制系统的工程技术人员参考。

目 录

前 言

| | |
|----------------------------|----|
| 第 1 章 概述 | 1 |
| 1.1 实时系统的定义..... | 1 |
| 1.1.1 实时..... | 1 |
| 1.1.2 实时系统..... | 2 |
| 1.1.3 实时计算机系统..... | 3 |
| 1.2 实时系统的功能要求..... | 4 |
| 1.2.1 数据收集..... | 4 |
| 1.2.2 数字控制..... | 6 |
| 1.2.3 人机互动..... | 6 |
| 1.3 实时系统的时间要求..... | 6 |
| 1.3.1 控制回路的时间要求..... | 7 |
| 1.3.2 延迟抖动最小化..... | 10 |
| 1.3.3 错误检测延迟最小化..... | 10 |
| 1.4 可依赖性要求..... | 10 |
| 1.4.1 可靠性..... | 11 |
| 1.4.2 可维护性..... | 11 |
| 1.4.3 有效性..... | 11 |
| 1.4.4 安全性..... | 12 |
| 1.4.5 防护性..... | 13 |
| 1.5 实时系统的分类方法..... | 13 |
| 1.5.1 根据应用特性分类..... | 13 |
| 1.5.2 根据计算机应用的设计和实现分类..... | 16 |
| 1.6 实时系统的应用前景..... | 17 |
| 1.6.1 工厂自动化系统..... | 17 |

| | |
|----------------------|-----------|
| 1.6.2 嵌入式实时系统 | 18 |
| 1.6.3 多媒体系统 | 21 |
| 习题 | 22 |
| 第 2 章 实时系统模型化 | 24 |
| 2.1 合理的抽象 | 24 |
| 2.1.1 抽象的定义 | 24 |
| 2.1.2 模型化的目的 | 25 |
| 2.1.3 假设覆盖率 | 25 |
| 2.1.4 相关属性 | 26 |
| 2.1.5 无关细节 | 28 |
| 2.1.6 系统架构 | 29 |
| 2.2 任务 | 29 |
| 2.2.1 任务的分类 | 30 |
| 2.2.2 任务的逻辑控制与时间控制 | 30 |
| 2.2.3 任务的事件触发与时间触发 | 33 |
| 2.2.4 任务的最坏情况执行时间 | 34 |
| 2.3 状态 | 38 |
| 2.3.1 状态的定义 | 38 |
| 2.3.2 基态 | 40 |
| 2.3.3 数据库组件 | 41 |
| 2.4 报文 | 41 |
| 2.4.1 报文的概念 | 41 |
| 2.4.2 报文结构 | 42 |
| 2.4.3 事件信息与状态信息 | 43 |
| 2.4.4 事件触发报文 | 44 |
| 2.4.5 时间触发报文 | 44 |
| 2.5 节点 | 45 |
| 2.5.1 接口特征 | 45 |
| 2.5.2 链接接口 | 47 |
| 2.5.3 本地接口 | 47 |
| 2.5.4 技术独立接口 | 48 |
| 2.5.5 技术依赖接口 | 48 |

| | | |
|--------------|------------------|-----------|
| 2.5.6 | 运行接口 | 48 |
| 2.5.7 | 网关 | 50 |
| 2.6 | 链接接口规范 | 51 |
| 2.6.1 | 报文传输规范 | 51 |
| 2.6.2 | 报文操作规范 | 52 |
| 2.6.3 | 报文元级规范 | 53 |
| 2.7 | 节点集成 | 54 |
| 2.7.1 | 可组性原则 | 54 |
| 2.7.2 | 分级集成 | 55 |
| 2.7.3 | 系统的系统 | 56 |
| | 习题 | 58 |
| 第 3 章 | 全局时钟同步 | 61 |
| 3.1 | 时间与顺序 | 61 |
| 3.1.1 | 顺序及其分类 | 61 |
| 3.1.2 | 时钟 | 63 |
| 3.1.3 | 时钟精密度与时钟准确度 | 65 |
| 3.1.4 | 时间标准 | 66 |
| 3.2 | 时间测量 | 67 |
| 3.2.1 | 全局时间 | 68 |
| 3.2.2 | 时间间隔测量 | 69 |
| 3.2.3 | π/Δ -领先 | 70 |
| 3.2.4 | 时间测量的基本限制 | 71 |
| 3.3 | 密集时基与稀疏时基 | 71 |
| 3.3.1 | 密集时基 | 72 |
| 3.3.2 | 稀疏时基 | 72 |
| 3.3.3 | 时空点阵 | 73 |
| 3.3.4 | 时间的循环表示形式 | 74 |
| 3.4 | 内部时钟同步 | 74 |
| 3.4.1 | 同步条件 | 75 |
| 3.4.2 | 中央主节点同步算法 | 76 |
| 3.4.3 | 分布式容错同步算法 | 77 |
| 3.4.4 | 状态修正与速率修正 | 81 |

| | | |
|------------|-------------------|-----------|
| 3.5 | 外部时钟同步 | 82 |
| 3.5.1 | 运行原理 | 82 |
| 3.5.2 | 时间格式 | 83 |
| 3.5.3 | 时间网关 | 84 |
| 3.6 | FlexRay系统的分布式时钟同步 | 84 |
| 3.6.1 | 时间表示形式 | 84 |
| 3.6.2 | 同步进程 | 85 |
| 3.6.3 | 时间偏差测量 | 87 |
| 3.6.4 | 修正值计算 | 87 |
| 3.6.5 | 时钟修正 | 89 |
| | 习题 | 90 |
| 第4章 | 实时实体与映像 | 92 |
| 4.1 | 实时实体 | 92 |
| 4.1.1 | 实时实体的控制范围 | 92 |
| 4.1.2 | 离散与连续实时实体 | 93 |
| 4.1.3 | 实时实体的观测 | 94 |
| 4.2 | 实时映像与实时对象 | 95 |
| 4.2.1 | 实时映像 | 96 |
| 4.2.2 | 实时对象 | 96 |
| 4.2.3 | 时间准确性 | 96 |
| 4.2.4 | 实时映像的分类 | 99 |
| 4.2.5 | 状态估计 | 101 |
| 4.3 | 持久性和幂等性 | 102 |
| 4.3.1 | 持久性 | 102 |
| 4.3.2 | 幂等性 | 104 |
| 4.4 | 确定性 | 105 |
| 4.4.1 | 确定性的定义 | 105 |
| 4.4.2 | 初始状态一致 | 107 |
| 4.4.3 | 非确定性设计结构 | 107 |
| 4.4.4 | 确定性恢复 | 108 |
| 4.5 | 信号的时间和约定 | 109 |
| 4.5.1 | 时间的双重作用 | 110 |

| | | |
|-------|---------------|-----|
| 4.5.2 | 数据及其语法约定和语义约定 | 111 |
| 4.6 | 过程I/O | 113 |
| 4.6.1 | 模拟I/O | 113 |
| 4.6.2 | 数字I/O | 114 |
| 4.6.3 | 探询采样机制 | 114 |
| 4.6.4 | 中断采样机制 | 116 |
| 4.6.5 | 容错执行器 | 118 |
| 4.6.6 | 智能仪表 | 120 |
| 4.6.7 | 紧凑组件 | 120 |
| | 习题 | 121 |
| 第5章 | 容错 | 123 |
| 5.1 | 故障、错误、失效和异常 | 123 |
| 5.1.1 | 故障 | 123 |
| 5.1.2 | 错误 | 125 |
| 5.1.3 | 失效 | 126 |
| 5.1.4 | 异常 | 128 |
| 5.1.5 | 系统容错与具体应用容错 | 128 |
| 5.2 | 错误、失效与异常检测 | 129 |
| 5.2.1 | 错误检测 | 129 |
| 5.2.2 | 失效检测 | 131 |
| 5.2.3 | 异常检测 | 131 |
| 5.3 | 失效单元 | 132 |
| 5.3.1 | 故障假设 | 133 |
| 5.3.2 | 节点作为失效单元 | 135 |
| 5.4 | 容错单元 | 137 |
| 5.4.1 | 故障静默FCU | 137 |
| 5.4.2 | 三模冗余 | 138 |
| 5.4.3 | 拜占庭弹性容错单元 | 140 |
| 5.4.4 | 成员资格服务 | 141 |
| 5.4.5 | 鲁棒系统结构 | 142 |
| 5.5 | 设计多样性 | 143 |
| 5.5.1 | 软件版本多样化 | 143 |

| | | |
|--------------|-----------------|------------|
| 5.5.2 | 系统层次化 | 145 |
| 5.6 | 修复节点的恢复 | 146 |
| 5.6.1 | 恢复点 | 146 |
| 5.6.2 | 基态最小化 | 147 |
| 5.6.3 | 节点重启 | 147 |
| | 习题 | 148 |
| 第 6 章 | 实时通信 | 150 |
| 6.1 | 实时通信的要求 | 150 |
| 6.1.1 | 时效性 | 150 |
| 6.1.2 | 通信可依赖性 | 151 |
| 6.1.3 | 灵活性 | 153 |
| 6.1.4 | 实时通信系统的物理结构 | 153 |
| 6.2 | 实时通信设计问题 | 154 |
| 6.2.1 | 细腰通信模型 | 155 |
| 6.2.2 | 物理性能限制 | 156 |
| 6.2.3 | 流量控制 | 158 |
| 6.2.4 | 猛烈摆动 | 160 |
| 6.3 | 事件触发通信 | 161 |
| 6.3.1 | 以太网 | 162 |
| 6.3.2 | 控制器局域网 | 167 |
| 6.3.3 | 用户数据报协议 | 169 |
| 6.4 | 速率受限通信 | 171 |
| 6.4.1 | 令牌总线协议 | 171 |
| 6.4.2 | 微时隙协议 ARINC 629 | 180 |
| 6.4.3 | 航空电子全双工交换式以太网 | 181 |
| 6.4.4 | 音频/视频总线 | 182 |
| 6.5 | 时间触发通信 | 183 |
| 6.5.1 | FlexRay | 184 |
| 6.5.2 | 时间触发协议 | 190 |
| 6.5.3 | 时间触发以太网 | 192 |
| 6.6 | 物理层 | 192 |
| 6.6.1 | 异步/同步通信的特征 | 193 |

| | | |
|------------|-----------------------|------------|
| 6.6.2 | 数字数据的传输编码 | 193 |
| 6.6.3 | 特征元形状 | 194 |
| 6.6.4 | 网络拓扑 | 194 |
| | 习题 | 195 |
| 第7章 | 实时操作系统 | 197 |
| 7.1 | 循环调度程序与抢占式多任务操作系统的差异 | 198 |
| 7.1.1 | 简单循环调度及其特点 | 198 |
| 7.1.2 | 抢占式调度的概念及其特点 | 202 |
| 7.1.3 | 使用实时操作系统的原因 | 203 |
| 7.2 | 实时操作系统的基本概念 | 204 |
| 7.2.1 | 实时操作系统的定义 | 204 |
| 7.2.2 | 实时操作系统任务及其状态 | 205 |
| 7.2.3 | 实时操作系统的任务控制块 | 207 |
| 7.2.4 | 任务到任务的上下文切换 | 208 |
| 7.2.5 | 可重入代码 | 209 |
| 7.2.6 | 资源与任务同步 | 210 |
| 7.3 | 任务管理 | 210 |
| 7.3.1 | 时间触发系统的任务管理 | 210 |
| 7.3.2 | 事件触发系统的任务管理 | 211 |
| 7.4 | 同步、互斥与通信 | 212 |
| 7.4.1 | 任务间同步与通信 | 213 |
| 7.4.2 | 节点间通信 | 215 |
| 7.5 | 中断与时间管理 | 217 |
| 7.5.1 | 中断处理 | 217 |
| 7.5.2 | 时间管理 | 219 |
| 7.6 | 错误检测方法 | 220 |
| 7.7 | 实时操作系统实例——OSEK | 221 |
| 7.7.1 | OSEK OS | 222 |
| 7.7.2 | OSEK实现语言 | 228 |
| 7.7.3 | AUTOSAR OS对OSEK OS的扩展 | 231 |
| 7.7.4 | OSEK COM | 231 |
| 7.7.5 | OSEK NM | 234 |

| | |
|---------------------|------------|
| 习题 | 235 |
| 第8章 实时调度 | 237 |
| 8.1 实时调度问题 | 237 |
| 8.1.1 实时调度算法的分类 | 237 |
| 8.1.2 可调度性分析 | 240 |
| 8.2 静态调度 | 246 |
| 8.2.1 时间在静态调度中的作用 | 247 |
| 8.2.2 搜索树在静态调度中的应用 | 247 |
| 8.2.3 静态调度表灵活性的增强方法 | 248 |
| 8.3 动态调度 | 250 |
| 8.3.1 独立任务的调度 | 250 |
| 8.3.2 非独立任务的调度 | 252 |
| 8.4 其他调度策略 | 256 |
| 8.4.1 分布式系统的调度问题 | 257 |
| 8.4.2 反馈调度 | 257 |
| 习题 | 258 |
| 第9章 系统设计 | 260 |
| 9.1 系统设计过程 | 260 |
| 9.1.1 设计问题 | 261 |
| 9.1.2 系统设计步骤 | 263 |
| 9.2 系统设计形式 | 265 |
| 9.2.1 模型化设计 | 265 |
| 9.2.2 节点化设计 | 266 |
| 9.2.3 架构设计语言 | 267 |
| 9.2.4 分解测试 | 268 |
| 9.2.5 典型开发流程 | 269 |
| 9.3 安全关键性系统设计 | 271 |
| 9.3.1 安全性的定义 | 271 |
| 9.3.2 安全性分析 | 272 |
| 9.3.3 安全案例 | 273 |
| 9.3.4 安全标准 | 276 |

| | |
|-----------------------------|------------|
| 9.4 系统安全性分析方法 | 278 |
| 9.4.1 故障树分析法 | 278 |
| 9.4.2 失效模式及影响分析 | 286 |
| 9.4.3 可依赖性的模型化 | 287 |
| 9.5 可维护性设计 | 287 |
| 9.5.1 维护成本 | 288 |
| 9.5.2 维护策略 | 288 |
| 9.5.3 软件维护 | 289 |
| 9.6 实时架构项目 | 290 |
| 9.6.1 SPRING系统 | 290 |
| 9.6.2 容错多机架构 | 291 |
| 9.6.3 时间触发架构 | 292 |
| 习题 | 297 |
| 第10章 系统评估 | 299 |
| 10.1 测试 | 299 |
| 10.2 节点化系统测试 | 304 |
| 10.3 正式方法 | 306 |
| 10.4 故障注入技术 | 308 |
| 10.4.1 物理故障注入 | 308 |
| 10.4.2 软件故障注入 | 311 |
| 10.4.3 传感器和执行器失效 | 312 |
| 习题 | 312 |
| 附录A 缩写词 | 313 |
| 附录B 故障树结构单元与符号 | 317 |
| 参考文献 | 319 |

第 1 章 概 述

最近十多年里,嵌入式实时系统、工业自动化系统和多媒体系统的应用强劲增长,极大地推动了实时系统概念的形成和发展。本章从实时系统的定义出发,探讨实时系统的基本组成,以及系统在功能、时间和可依赖性方面的要求,并根据控制应用的特点,重点讲述在时间方面的概念和要求。实时系统的分类方法有多种,这里特别强调了弱实时(*soft real time*)系统和强实时(*hard real time*)系统之间的基本区别。本章还从经济角度分析实时系统的应用前景。

1.1 实时系统的定义

在介绍实时系统的定义之前,首先回顾一下“实时”和“系统”这两个术语。“实时”是一个不太容易理解的术语,许多人想当然地认为实时就是快。实际上,实时表示具有确定性的响应,即在给定的时间周期内,对某个事件做出可靠、准确的响应的能力。“系统”是指由相互制约的若干部分所构成的具有特定功能的整体。系统的状态由描述系统行为特征的变量来表示。随着时间的推移,系统会不断演化。外部环境的影响、内部组成的相互作用和人为的控制作用等,是导致系统状态和演化进程发生变化的主要因素。

很多文献给出了实时系统的定义,然而不同文献给出的定义不尽相同,至今也没有一个被人们广泛接受的定义。下面将从控制应用的角度,更加详细地讲述实时和实时系统的相关概念。

1.1.1 实时

简单地说,实时是用来描述实际应用的定时要求的。不同应用和不同用户的定时要求不尽相同,因此事物的实时性没有一个统一的时间限制。为了更加精确地定义系统的实时性能,经常使用弱实时和强实时两个术语。

实施弱实时的系统可以有不同的响应速率,而且不会影响整个系统的整体功能。例如,在温度监视系统中,温度不会快速改变,获取数据的速率相对较慢,可以每秒读数一次,读数间隔的轻微变化不会影响整个系统的功能。

然而,强实时的要求与前者不同,在一个绝对的时间内,它的响应速率必须是无差错的、准确的。例如,蒸馏过程的控制,必须以指定的时间间隔一致地采集压

力信号,以便及时做出开、关压力阀门的重要决策。如果不能在指定的时间周期内执行控制回路,那么压力可能增大到危险的程度。

1.1.2 实时系统

系统行为的正确性不仅取决于计算的逻辑结果,而且与产生这些结果的物理时间有关,这类系统称为实时系统^[1]。其中,系统行为表示系统随时间推移的输出序列。实时计算机系统属于实时系统的一部分。

实时系统是随时间变化的。例如,在化学反应系统中,即使计算机控制系统已经停止运行,化学反应仍将继续改变状态。因此,将实时系统分解成一组自成体系(self-contained)的子系统是合理的,这种子系统通常称为簇。图1-1给出了一个实时系统实例,整个系统被分解成被控对象(被控簇)、实时计算机系统(计算簇)和操作员(操作员簇)三个子系统。

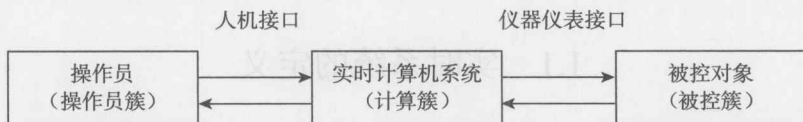


图1-1 实时系统实例

计算簇一般是分布式的,系统中的节点不只一个,各个节点通过实时通信网络相互连接,如图1-2所示,图中的A~F节点都是计算机节点。被控簇可以是物理设备或机器。操作员簇是指人机交互中的人类因素,但这里仅考虑操作员与计算簇之间的互动模式,并不关心操作终端的信息表示形式。通常,把被控簇和操作员簇统称为计算簇(实时计算机系统)的环境。

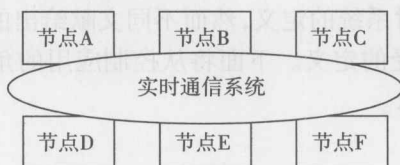


图1-2 分布式计算机系统

操作员与实时计算机系统之间的接口称为人机接口,被控对象与实时计算机系统之间的接口称为仪器仪表接口。人机接口是由输入设备(如键盘、鼠标)和输出设备(如显示器)组成的,用于实现与操作员的连接。仪器仪表接口是由传感器和执行器组成的,用于将被控簇中的物理信号(如电压、电流)变换成数字信号,或者将数字信号变换成被控簇中的物理信号。拥有仪器仪表接口的节点称为接口节点。

1.1.3 实时计算机系统

用从过去到未来的有向直线表示时间的进展,将时间线上的切口称为时刻(instant),发生在某一时刻的任何典型情况称为事件(event),描述事件的信息称为事件信息。当前时间点(现在)是一个非常特殊的事件,它将过去和未来分离开来(该时间模型建立在牛顿物理学基础上,忽略了相对论效应)。时间线上的间隔是由其起始事件和终止事件定义的,间隔的持续时间等于终止事件的时间减去起始事件的时间。数字时钟把时间线分割成一系列等长的持续时间,该等长的持续时间称为时钟粒度(granule),它们被特殊的周期性事件分隔开来,这些时间线上的特殊周期性事件称为时钟节拍(tick)。

实时计算机系统必须在指定的时间间隔内,对来自环境(被控簇或操作员簇)的激励做出反应,这里的指定时间间隔是由环境决定的,实时计算机系统必须产生结果的时刻称为截止时间(deadline)。如果截止时间已过,而产生的结果仍然有用,那么这个截止时间被定为弱截止时间。如果错过截止时间可能导致严重后果,那么该截止时间被定义为强截止时间。例如,一个有信号灯的铁路和公路交叉口,若信号灯没有在火车到达前变成“红”色,则可能导致意外事故。必须满足至少一个强截止时间的实时计算机系统称为强实时计算机系统,或安全关键性实时计算机系统。若系统不存在必须满足的强截止时间,则该系统称为弱实时计算机系统。

综上,实时计算机系统必须对给定的输入做出响应,而且强实时计算机系统的响应一定要在指定的截止时间之前完成。图1-3给出了实时计算机系统的时间概念模型^[2]。

图1-3中,事件可能来自外界输入,如传感器的输入、硬件定时器中断等。一般情况下,系统要对事件做出响应,该响应可能会、也可能不会输出至外界。响应通常要在指定的截止时间之前发生。例如,如果给定输入是车轮的转动,那么截止时间是车轮转动一周的时间。从图中可以看出,事件发生后到开始处理之前有一段时间延迟(latency),产生响应的时间,除了包含该延迟,还包括一定的处理时间。一个实时计算机系统通常需要处理大量的事件。

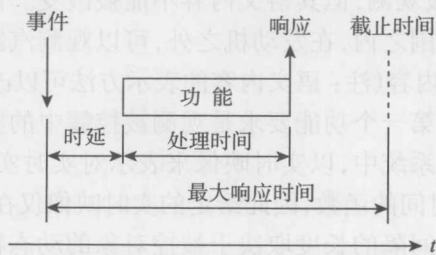


图1-3 实时计算机系统的时间概念模型