

图书基本信息

书名：<<嵌入式实时操作系统 μ C/OS原理与实践>>

13位ISBN编号：9787121154416

10位ISBN编号：7121154412

出版时间：2012-2

出版时间：电子工业出版社

作者：卢有亮

页数：288

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>

内容概要

本书内容包括：实时操作系统基础、任务管理、中断和时间管理、事件管理、消息管理、内存管理、移植等。

本书内容翔实，图文并茂，采用逐步深入、反复印证的方法，从数据结构的设计入手，再到代码分析、示例验证的剖析方法，逐层深入讲解，给出在虚拟平台下的移植示例和针对各章内容示例，并给出了基于NIO

II的FPGA系统上移植的例子。

本书适用于计算机、电子、通信、自动化及相关专业大学本科、研究生，也适用于广大嵌入式开发工程师技术人员、电子技术研究人员、操作系统研究人员。

书籍目录

Contents

第1章 实时操作系统基础

1.1 操作系统概述

1.1.1 什么是操作系统

1.1.2 操作系统基本功能

1.2 实时操作系统概述

1.2.1 什么是实时操作系统

1.2.2 实时操作系统的基本特征

1.3 任务

1.3.1 任务简介

1.3.2 多任务

1.3.3 任务状态

1.3.4 任务切换

1.3.5 可重入函数和不可重入函数

1.4 基于优先级的不可剥夺内核

1.4.1 内核

1.4.2 基于优先级的调度算法

1.4.3 不可剥夺型内核和可剥夺型内核

1.5 同步与通信

1.5.1 同步

1.5.2 互斥

1.5.3 临界区

1.5.4 任务事件

1.5.5 信号量

1.5.6 互斥信号量

1.5.7 事件标志组

1.5.8 消息邮箱和消息队列

1.6 中断和时钟

1.7 内存管理

习题

第2章 任务管理

2.1 任务管理数据结构

2.1.1 任务控制块

2.1.2 空闲链表和就绪链表

2.1.3 任务优先级指针表

2.1.4 任务堆栈

2.1.5 任务就绪表和就绪组

2.2 任务控制块初始化

2.2.1 代码解析

2.2.2 流程分析

2.3 操作系统初始化

2.3.1 代码解析

2.3.2 流程分析

2.4 任务的创建

2.4.1 OSTaskCreate代码解析

- 2.4.2 OSTaskCreate流程分析
- 2.4.3 OSTaskCreateExt代码解析
- 2.4.4 OSTaskCreateExt流程分析
- 2.5 任务的删除
 - 2.5.1 任务删除代码解析
 - 2.5.2 任务删除流程分析
 - 2.5.3 请求删除任务代码解析
 - 2.5.4 请求删除任务流程
- 2.6 任务挂起和恢复
 - 2.6.1 OSTaskSuspend代码解析
 - 2.6.2 OSTaskSuspend流程分析
 - 2.6.3 OSTaskResume代码解析
 - 2.6.4 OSTaskResume流程分析
- 2.7 任务的调度和多任务的启动
 - 2.7.1 任务调度器
 - 2.7.2 任务切换函数
 - 2.7.3 中断中的任务调度
 - 2.7.4 多任务的启动
- 2.8 特殊任务
 - 2.8.1 空闲任务OS_TaskIdle
 - 2.8.2 统计任务OS_TaskStat
- 2.9 任务管理总结
- 习题
- 第3章 中断和时间管理
 - 3.1 中断管理
 - 3.1.1 中断管理核心思路
 - 3.1.2 中断处理的流程
 - 3.1.3 时钟中断服务
 - 3.2 时间管理
 - 3.2.1 时间管理主要数据结构
 - 3.2.2 时间的获取和设置
 - 3.2.3 任务延时函数OSTimeDly
 - 3.2.4 任务按分秒延迟函数OSTimeDlyHMSM
 - 3.2.5 延时恢复函数OSTimeDlyResume
- 习题
- 第4章 事件管理
 - 4.1 事件管理的重要数据结构
 - 4.1.1 事件控制块 (ECB)
 - 4.1.2 事件等待组和事件等待表
 - 4.1.3 事件控制块空闲链表
 - 4.2 事件管理程序
 - 4.2.1 事件控制块 (ECB) 初始化
 - 4.2.2 事件等待表初始化
 - 4.2.3 设置事件等待
 - 4.2.4 取消事件等待
 - 4.2.5 将等待事件的任务就绪
 - 4.3 信号量管理

- 4.3.1 信号量的建立OSSemCreate
- 4.3.2 信号量的删除OSSemDel
- 4.3.3 请求信号量OSSemPend
- 4.3.4 提交信号量
- 4.3.5 无等待请求信号量
- 4.3.6 放弃等待信号量
- 4.3.7 信号量值设置
- 4.3.8 查询信号量状态
- 4.3.9 信号量应用举例
- 4.4 互斥信号量管理
 - 4.4.1 互斥信号量的建立
 - 4.4.2 请求互斥信号量
 - 4.4.3 互斥信号量的删除
 - 4.4.4 发互斥信号量
 - 4.4.5 无等待请求互斥信号量
 - 4.4.6 查询互斥信号量状态
 - 4.4.7 改变任务的优先级并重新就绪
 - 4.4.8 互斥信号量应用举例
- 4.5 事件标志组管理
 - 4.5.1 事件标志组数据结构
 - 4.5.2 事件标志组初始化
 - 4.5.3 创建事件标志组
 - 4.5.4 事件标志组阻塞函数
 - 4.5.5 请求事件标志
 - 4.5.6 删除事件标志组
 - 4.5.7 提交事件标志组
 - 4.5.8 标志节点任务就绪
 - 4.5.9 无等待的请求事件标志
 - 4.5.10 事件标志管理应用举例

习题

第5章 消息管理

- 5.1 消息邮箱
 - 5.1.1 建立消息邮箱
 - 5.1.2 等待消息
 - 5.1.3 发消息
 - 5.1.4 删除消息邮箱
 - 5.1.5 放弃等待邮箱
 - 5.1.6 无等待请求消息
 - 5.1.7 查询消息邮箱状态
 - 5.1.8 消息邮箱的例子
- 5.2 消息队列
 - 5.2.1 消息队列数据结构
 - 5.2.2 初始化消息队列
 - 5.2.3 建立消息队列
 - 5.2.4 发消息到消息队列
 - 5.2.5 等待消息队列中的消息
 - 5.2.6 删除消息队列

5.2.7 取得消息队列的状态

5.2.8 消息队列应用举例

习题

第6章 内存管理

6.1 内存管理数据结构

6.1.1 内存控制块

6.1.2 内存控制块实体

6.1.3 空闲内存控制块链表

6.1.4 内存分区

6.2 内存控制块初始化

6.3 创建内存分区

6.4 内存分区获取

6.5 内存分区释放

6.6 查询内存分区的状态

6.7 内存管理实例

习题

第7章 移植

7.1 移植说明

7.1.1 μ C/OS-II的代码结构

7.1.2 操作系统中与CPU相关的代码解析

7.1.3 μ C/OS-II移植步骤

7.2 在VISUAL C++6.0上实现基于WINDOWS的虚拟 C/OS-II移植

7.2.1 目录结构和工程的建立

7.2.2 包含文件includes.h

7.2.3 os_cpu.h中修改的代码

7.2.4 os_cpu.c中修改的代码

7.2.5 主程序代码实现

7.2.6 移植测试

7.3 在基于NIOS软核的FPGA嵌入式系统下的 C/OS-II移植

7.3.1 系统结构

7.3.2 NIOS-II寄存器

7.3.3 os_cpu.h的移植代码

7.3.4 os_cpu.c的移植代码

7.3.5 os_cpu.s的移植代码

7.3.6 工程的创建和移植测试

习题

章节摘录

版权页：插图：互斥信号量是一种特殊的信号量，这不仅在于该信号量只有用于互斥资源的访问，还在于使用互斥信号量管理需要解决的优先级反转问题。

假如系统中有3个任务，分别是高优先级、中优先级和低优先级，当低优先级的任务在运行的时候访问互斥资源，而中优先级的任务运行时将使低优先级的任务得不到运行而死抱着资源不放。这时，高优先级的任务开始运行的时候，必须等待中优先级的任务运行完成，然后等低优先级的任务访问资源完成才行。

如果在低优先级的任务访问资源过程中又有中优先级任务运行，那么高优先级的任务只有继续等待。这种情况就是优先级反转。

在 μ C/OS-II对互斥信号量的管理中，针对这个问题采用了优先级继承机制。

优先级继承机制是一种对占用资源的任务的优先级进行升级的机制，用以优化系统的调度。

例如，当前的任务的优先级是比较低的。

高优先级的任务请求互斥信号量的时候因为信号量已被占有，所以只有阻塞。

这时有中优先级的任务就绪。

如果不采用优先级继承，那么高优先级的任务是竞争不过中优先级的任务的。

采用优先级继承机制，将占有资源的低优先级的任务临时设置为一个很高的优先级，允许其在占有资源的时候临时获得特权，先于中优先级任务完成，在访问互斥资源结束又回到原来的优先级，这样高优先级的任务就会先于中优先级的任务运行，解决了这个问题。

本书在第4章的“4.4互斥信号量管理”一节还将详细论述该问题并给出例程。

编辑推荐

《嵌入式实时操作系统 μ C/OS原理与实践》 μ C/OS是高实时性、多任务的操作系统，也是源代码对非商业用途开放的操作系统，笔者结合自己的实际工作经验，通过本书详细阐释 μ C/OS的系统原理及实战技巧。

《嵌入式实时操作系统 μ C/OS原理与实践》对最新版本的 μ C/OSII-2.91进行深入剖析，从实时操作系统的原理开始，对重要的数据结构和代码结合相关实例进行详细解析，以便于读者掌握。通过《嵌入式实时操作系统 μ C/OS原理与实践》的学习，能够帮助读者全面、深入地掌握嵌入式实时操作系统的基本原理，全方位接触 μ C/OS，对所学知识融会贯通，能够灵活应用到实际的项目当中。

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>