

<<VxWorks设备驱动开发详解>>

图书基本信息

书名：<<VxWorks设备驱动开发详解>>

13位ISBN编号：9787121128288

10位ISBN编号：7121128284

出版时间：2011-3

出版时间：电子工业出版社

作者：曹桂平 等编著

页数：409

字数：672000

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>

<<VxWorks设备驱动开发详解>>

内容概要

本书结合作者实际的开发经验，对VxWorks操作系统内部的机制及其各类设备驱动的开发进行了深入的讲解和分析。

全书分3篇共11章，第1篇对VxWorks操作系统的主要组件如任务、任务调度、任务间通信、内存管理、中断处理进行了较为细致的分析；其后对很多VxWorks开发者不甚了解的VxWorks内核映像类型以及启动方式和流程进行了详细的说明和解析。

第2篇是驱动开发的准备阶段，着重介绍了VxWorks系统下设备驱动的内核层次结构。

第3篇作为本书的重点，每章对应一类驱动，结合开发实例，详细而完整地分析了VxWorks下普通字符设备驱动、串口驱动、普通块设备驱动、Flash设备驱动、网络设备驱动以及USB设备驱动的设计和实现。

读者对象：本书面向广大工程技术工作者，既可作为高等院校教师和相关专业学生的教材，又可作为各类培训班的培训教程。

<<VxWorks设备驱动开发详解>>

书籍目录

第1篇 VxWorks操作系统快速入门篇

第1章 VxWorks嵌入式操作系统的特点与应用

1.1 嵌入式系统概述

1.1.1 嵌入式系统定义

1.1.2 嵌入式系统组成和特点

1.1.3 嵌入式系统发展趋势

1.1.4 实时操作系统

1.2 VxWorks操作系统基本特点

1.2.1 操作系统内核结构——微内核和宏内核

1.2.2 VxWorks内核——高性能的微内核设计

1.2.3 VxWorks开发支持——可裁减的运行软件

1.2.4 VxWorks网络支持——综合的网络工具

1.2.5 VxWorks移植性支持

1.2.6 VxWorks操作系统选件

? 1.3 VxWorks操作系统应用范围

1.4 本章小结

第2章 VxWorks操作系统的基本组件

2.1 VxWorks任务

2.1.1 内核实现基本原理

2.1.2 任务操作函数

2.1.3 深入了解任务栈

2.1.4 任务名长度问题

2.1.5 正确结束任务

2.1.6 任务的钩子函数——黑客机制

2.1.7 任务小结

2.2 VxWorks任务调度算法——基于优先级的抢占式调度

2.3 VxWorks任务间通信策略

2.3.1 信号量

2.3.2 消息队列

2.3.3 管道

2.3.4 网络套接字Socket

2.3.5 任务间通信的特殊机制：信号

2.4 VxWorks内存管理——虚拟地址空间支持

2.5 VxWorks中断处理——多层次的中断转移

2.5.1 VxWorks下中断转移过程详解（基于ARM平台）

2.5.2 中断上下文中为何不可调用可引起睡眠的函数

2.6 本章小结

第3章 VxWorks操作系统启动过程详解

3.1 VxWorks操作系统启动

3.1.1 VxWorks基本启动方式

3.1.2 VxWorks操作系统内存布局

3.1.3 下载型启动方式概述

3.1.4 ROM型启动方式概述

3.2 深入理解bootrom——下载启动方式下的“瑞士军刀”

3.2.1 bootrom的构成

<<VxWorks设备驱动开发详解>>

3.2.2 bootrom脚本的创建

3.2.3 脚本运行过程分析

3.2.4 bootrom的重定位

3.2.5 RAM中运行的bootrom代码

3.2.6 在bootrom中添加用户代码

3.2.7 其他注意事项及说明

3.3 深入VxWorks启动过程

3.3.1 ROM型映像早期启动流程详解

3.3.2 下载型映像早期启动流程详解

3.3.3 公共启动流程详解

3.4 VxWorks BSP文件组成

3.4.1 源文件

3.4.2 头文件

3.4.3 Makefile文件

3.4.4 扩展文件

3.4.5 说明文件

3.5 本章小结

第2篇 VxWorks设备驱动起步篇

第3篇 VxWorks之六大核心设备驱动

参考文献

章节摘录

信号量的底层实现可以简单地看做是一个内核维护的全局变量，对于用于互斥机制的信号量，这个内核全局变量初始化为1，当一个任务需要访问该信号量保护的资源时，其首先检查这个内核全局变量的值是否为1，如非1，则表示已存在其他任务在使用资源，就等待；如为1，表示资源当前可被访问，则这个任务首先将这个内核全局变量的值设置为0，阻止其他任务的访问，而自身就可以安全地使用该资源。

此处有一个漏洞是，在当前任务修改内核全局变量的同时，另一个任务可能同时在检查这个全局变量的值，很可能造成另一个任务检查到全局变量值为1后，当前任务才完成全局变量0值的设置，此时就有两个任务在使用资源，造成内核状态的不一致，极端情况下，将造成整个系统的崩溃。

内核对这种情况进行了特殊处理，一般是将变量的改变操作作为一个原子操作（如x86下提供的Lock指令）完成。

这也是内核提供的任务间通信机制和用户层任务间通信机制的根本区别：内核提供的机制已经从根本上保证了足够的安全性。

基于各种资源不同的使用方式，VxWorks信号量机制具体提供了三种信号量：通用信号量、互斥信号量、资源计数信号量。

通用信号量既可用于同步，也可用于资源计数，此时资源数通常为1（当资源数为1时，也可以称之为互斥）。

互斥信号量针对在使用过程中的一些具体问题（如优先级反转）做了优化，更好地服务于任务间互斥需求；资源计数信号量用于资源数较多，同时可供多个任务使用的场合。

2.3.2 消息队列 消息队列内核实现上实际是一个结构数组，数组大小和数组中元素的容量在创建消息队列时被确定。

在创建消息队列时指定的另外一个参数是消息队列满时任务等待基于的策略：FIFO或者优先级排序。消息队列是VxWorks内核提供的任务间传递较多信息的一种机制，不过这种机制存在很大的局限性，即每个消息的最大长度是固定的。

当然，在这个最大长度范围内从用户层而言是可变的，但是对于内核维护而言，所有的消息都具有相同的长度，因为无论实际消息的长度如何，内核都将按最大长度分配内存空间。

当然，如果对每个消息都采用动态内存分配方式，可以消除最大长度限制，但是这并不是VxWorks提供的消息机制。

VxWorks内核提供的消息机制在创建消息队列时必须指定单个消息的最大长度以及消息的数量，在消息队列成功创建后，这些参数都是固定不变的。

我们可以如此想象内核对于消息队列的实现，在消息队列创建之时，内核分配一个大小为单个消息最大长度与消息数量乘积的内存区域，可以将此看做是一个数组，数据元素个数为消息数量，每个元素的大小为单个消息最大长度。

当用户发送一个消息时，内核将消息内容存入数组中下一个空闲元素中，用户读取消息时，将读取数组中下一个非空元素，底层基本实现为一个环形缓冲区。

VxWorks最多只区分两个优先级的消息，对于高优先级的消息将从数组开始处存储，对于普通优先级的消息将从数组尾部开始存储，而读取时从数组头部开始读取，从而保证高优先级的消息优先被传递。

。

当然，以上只是一种简单的类比，有助于读者理解VxWorks内核对于消息队列的实现。

.....

<<VxWorks设备驱动开发详解>>

编辑推荐

华清远见嵌入式培训中心成立于2004年，作为中国嵌入式培训的领导者，ARM、微软、谷歌、红帽等权威知名企业授权培训中心，七年培训的学员数量已突破一万八千名。凭借多年的专注和努力，该培训中心先后两次蝉联2008-2010“中国本土嵌入式系统十佳企业”，并荣获“微软嵌入式合作伙伴卓越奖”，工信部颁发的“中国嵌入式人才服务特别奖”，中国计算机行业协会“中国IT产业最具影响力教育机构金奖”、“中国计算机行业发展成就·最具社会责任奖”等荣誉。

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>