

# VxWorks 嵌入式实时操作系统的原理和实现

王金辉

(中国电子科技集团公司第 54 研究所,河北 石家庄 050081)

**摘 要** 介绍了 VxWorks 作为嵌入式实时操作系统具有的功能和特点。分析了 VxWorks 的任务调度管理机制,如何创建、管理任务、任务状态的切换、任务的优先级等。分析了 VxWorks 的任务间通信的机制,如何创建邮箱、收发邮件等,以及消息传递采用的方式。介绍了 VxWorks 在实际应用中的工程实现步骤和调试过程,同时介绍了几个重要概念(bootRom Image、bsp 等)。

**关键词** VxWorks; 嵌入式实时系统; bootRom Image; bsp

**中图分类号** TP316.2 **文献标识码** A **文章编号** 1003-3106(2007)01-0062-03

## Theory and Realization of VxWorks OS

WANG Jin-hui

(The 54th Research Institute of CETC, Shijiazhuang Hebei 050081, China)

**Abstract** This paper introduces the feature of VxWorks as embed operating system, analyses mechanism of VxWorks task, how to create and manage tasks, and priority of task. It also analyses mechanism of communication between tasks, how to create mail box, how to receive and send mail, and the method to send message, as well as the step of realize in application and process of debug, and some important conceptions (bootRom Image, bsp).

**Key words** VxWorks; operation system of embedding; bootRom Image; bsp

## 0 引言

随着科学技术的不断快速发展和对事件处理实时性要求的提高,在目前的通信产品开发中,功能相对单一的单片机(如 C51)已经逐步被功能强大的嵌入式处理器代替。硬件工作量的减少使得一些功能必须由软件完成,而软件代码的执行,必须依托于运行稳定、功能丰富的操作系统。VxWorks 是美国 Wind River System 公司的产品,是一种嵌入式操作系统(RTOS),它以良好的可靠性和卓越的实时性广泛地应用在通信、军事及航空航天等领域中。

## 1 VxWorks 概述

VxWorks 是目前比较流行的嵌入式操作系统,依据其良好的持续发展能力、高性能的内核和友好的用户开发环境,在通信领域得到了广泛的应用。

VxWorks 具有一个高性能的操作系统内核 Wind,主要特点包括快速多任务切换、抢占式任务调度、任务间通信手段多样化等,同时任务切换时间短、中断时延小、网络流量大,与其他嵌入式实时操作系统相比有一定的优势。

VxWorks 的任务调度策略以抢占式调度为基础,同时也可以使用时间片轮转调度算法,这种调度策略能够及时地响应高优先级的任务,而同级任务则可以选择时间片轮转法使多个同优先级的任务并发执行。

VxWorks 采用中断处理与普通任务分别在不同的栈中处理的中断处理机制,使得中断的产生只会引发一些关键寄存器的存储而不会导致任务的上下文切换,从而减少了中断延迟。VxWorks 的中断处理程序只完成在最小时间内通知中断已经产生,而将中断引起的非实时处理放在其他任务中完成,使得对中断的反应非常迅速。

VxWorks 具有友好的开发调试环境 Tornado,便于操作、配置和应用程序的开发调试,并可以运行到多种主机上,支持 Unix、WindowsNT 等。在 Tornado 中还可以增加或裁减系统功能,使应用程序的大小得到保障。

VxWorks 具有很好的兼容性,是最早兼容 POSIX 1003.1b 标准的嵌入式实时操作系统之一,同时支持 ANSI C 标准。

收稿日期:2006-03-09

## 2 VxWorks 的系统功能

### 2.1 任务管理机制

VxWorks 是一个多任务环境,任何一个正在执行的程序都可称之为任务。

每一个任务都有自己的上下文。上下文保存在任务控制块中,是任务转为运行状态时要查看的 CPU 的环境资源和系统资源。一个任务的上下文包括:任务程序计数器、CPU 寄存器和浮点寄存器、动态变量和调用函数的栈、标准输入输出的 I/O 分配、延时的时间、时间片的大小、内核控制结构、信号处理、调试和性能监视。

在 VxWorks 中,任务有多种状态,内核根据应用程序的需求调度任务状态的转换。主要的状态包括:

READY:任务已经得到所有需要的资源,只等待 CPU 分配运行时间;

PEND:任务由于某种资源不可用而阻塞,阻塞的任务可以参加系统任务调度;

DELAY:任务由于系统要求延时而进入睡眠状态;

SUSPEND:  
主要是因为调试才进入的状态。

状态之间的转换如图 1 所示。

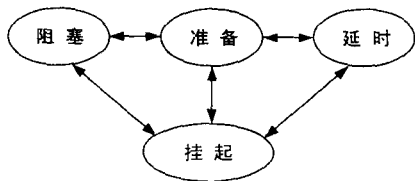


图 1 任务转换图

任务被创建后首先进入 Suspend 状态,根据某种系统调用在状态之间进行转换。如调用 TaskActive 就可以从 Suspend 转到 Ready 状态。任务初始化完成后处于 Suspend 状态。

在 VxWorks 中,基于优先级的抢占式调度算法是基本的任务调度机制,同时也可以选择时间片轮询算法。在抢占式调度机制中,每一个任务都有优先级,内核确保 CPU 分配给最高优先级的任务。抢占式调度机制的核心是当有更高优先级的任务准备运行时,内核保存正在执行的任务的上下文的同时立即切换到更高优先级任务的上下文。Wind 内核共有 256 个优先级,0 的优先级最高,255 的优先级

最低。任务在创建的时候优先级就已经被分配,在运行过程中可以调用系统功能修改任务的优先级。如果任务的优先级相同,可以采用时间片调度算法, CPU 均匀分配给每个任务相同的运行时间。

### 2.2 任务间的通信机制

VxWorks 提供了多种通信方式以满足各种形式的需求。

#### (1) 共享内存

适用于简单的数据共享。

#### (2) 信号量

在资源互斥和任务同步的情况下使用。VxWorks 的信号量提供了最快的任务间通信机制。信号量作为任务等待运行的条件或事件,一般在中断服务程序(ISR)和任务之间使用。首先创建一个空的二进制的信号量,然后在 ISR 中使用 SemGive 释放,在任务中使用 SemTake 得到,并对产生中断的事件进行分析和处理。

#### (3) 消息队列

在 VxWorks 中,主要的任务间通信使用消息队列。一般每个任务创建一个消息队列,消息队列使用 MsgQCreate 创建,其中的参数指定了可以容纳队列的个数和每个队列的长度。任务或 ISR 使用 MsgQSend 向指定的任务发送数据,任务使用 MsgQReceive 从指定的队列接收数据。如果没有数据,则任务进入阻塞状态并且被加到等待消息的任务队列中等待内核的调度。任务队列按照优先级或先进先出(FIFO)原则排序。消息的接收和发送都有超时参数,对于发送,如果目的队列没有足够空间,则在超时范围内等待。对于接收,如果接收队列没有消息,则在超时范围内等待。接收超时参数也可以设置为 WAIT - FOREVER,即如果没有消息则永远等待。

## 3 VxWorks 的工程实现

### 3.1 建立 BSP

板级支持包(Board Service Packet)的目的就是为 VxWorks 提供硬件环境的接口,加电后配置 CPU 的参数、初始化 RAM 和 FLASH、驱动串口和网口、加载 VxWorks 内核,建立最小的 VxWorks 系统。一般应用

在调试阶段,是程序开发必不可少的步骤和调试手段。

BSP 包括以下几个文件:

① Makefile: 包含了一些宏定义,包括:处理器类型、RAM 和 ROM (FLASH) 的大小、VxWorks 拷贝地址、文件目录等;

② RomInit.s: 屏蔽中断,初始化 CPU,关闭 cache,初始化 RAM;

③ Config.h: 包含所有的头文件和与 CPU 相关的宏定义,如对 cache 和 MMU 的配置,对 RAM 和 ROM 的定位以及大小配置,外部总线地址映射等;

④ BootConfig.c: 完成各种系统调用和硬件的初始化,如初始化网口、串口、时钟,启动 VxWorks 最小多任务环境,从宿主机上下载 VxWorks 镜像;

⑤ BootInit.c: 完成 flash 中程序到 RAM 中的拷贝。

目标板加电后,首先在 Flash 中执行 RomInit.s,完成对 CPU 的初始化,然后执行 BootInit.c,将代码从 Flash 拷贝到 RAM 中并执行各种基本 I/O 的初始化并从开发板下载 VxWorks。

BSP 执行的大多数操作是各种发布版本必须的操作,所以 BSP 的调试是 VxWorks 应用开发的基础。

BSP 开发的难点在于 RomInit.s 文件,因为这个文件中的代码不能跟踪,没有什么调试手段,但是一些重要的工作都在其中执行,如 RAM 的初始化、UPM 表的使用等。在通过串口或网口可以跟踪调试以前,点灯是最好的调试手段。

### 3.2 项目的建立和编译

通过编译环境 Tornado 可以建立 Project。建立时一般选择 Bootable 启动类型,然后依据调试成功的 BSP 建立项目。

项目建立后,可以通过 Tornado 裁减目标板不需要的系统组件,这样可以大大减小应用程序的长度。在 Tornado 中可以定义全局使用的宏定义和编译条件,然后生成需要的镜像(应用程序)。

### 3.3 VxWorks 镜像

镜像是最后的应用程序,依据具体的 BSP 和 Project 生成。根据不同的运行方式,可以分成 RAM 型和 ROM 型。

#### (1) VxWorks 镜像

属于 RAM 型,在 RAM 中运行,由 BootRom 通过 RS232 或 RJ45 从开发板或局域网下载到 RAM 中的 RAM\_LOW\_ADDR 处执行。这种形式的镜像一般在调试过程中使用,可以通过设置断点、任务监视等手段跟踪代码的运行,能方便快速地定位问题并解决问题。

#### (2) VxWorks\_rom 镜像

属于 RAM 型,在 RAM 中运行。应用程序固化在 Flash 或 Rom 中,加电后,拷贝至 RAM 中的 RAM\_HIGH\_ADDR 处执行。这种形式的镜像应用最普遍。但是如果一旦应用程序有问题,则必须重新烧录 Flash 或 Rom,程序升级时不方便。

#### (3) VxWorks\_romRes 镜像

属于 ROM 型。应用程序固化在 Flash 或 Rom 中并在 Flash 或 Rom 中运行。这种方式应用在 RAM 资源比较少的情况下。

通过调试成功的 BSP,使用串口或网口从主机上下载 VxWorks 镜像并运行。在运行后,可以使用 Target Server 对任务设置断点,查看变量、任务的状态等进行调试。

调试成功后,重新建立项目文件,添加代码文件,生成 VxWorks\_rom 镜像文件,然后通过 BDM 接口,使用 VisionClick 等烧录软件将 VxWorks\_rom 固化到 Flash 或 Rom 中。

## 4 结束语

依据对 VxWorks 嵌入式操作系统原理和实现方法的分析,以 Motorola 的 PowerPC 和 VxWorks 为核心,使用中断和邮箱通信机制并创建多个任务,研制了协议控制处理器,在实际的工程任务中收到了很好的效果。



#### 参考文献

- [1] 罗国庆. VxWorks 与嵌入式软件开发[M]. 北京:机械工业出版社,2004.
- [2] 陈智育. VxWorks 程序开发实践[M]. 北京:人民邮电出版社,2004.
- [3] 刘红梅. SCTP 在军事通信网络中的应用研究[J]. 无线电工程,2005,35(5):24-27.

#### 作者简介

王金辉 男,(1974-),中国电子科技集团第 54 研究所工程师。主要研究方向:通信技术。