

嵌入式操作系统中消息机制的研究

高睿*, 张明月

(辽宁广播电视大学 信息工程系, 辽宁 沈阳 110035)

摘要: 研究了实时的抢占式的小型嵌入式操作系统中消息机制的实现. 针对 Task 调度、Task 管理、同步通信、中断管理、时间管理机能这五部分进行了研究. 为保证多个相关任务在执行次序上进行协调, 阐述了同步通信机制所用三种方式 Event Flag、Semaphore、MailBox 的实现, 着重讲述的是同步通信中 Mail Box 的实现. 实际运行后证明系统完全满足通信需要.

关键词: 嵌入式系统; 消息机制; TASK; 邮箱.

中图分类号: TP29 文献标识码: A 文章编号: 1000-5846(2007)03-0241-03

多任务的实时系统中, 一项工作的完成往往要通过多个任务或多个任务与多个中断处理过程 (ISRs Interrupt Service Routine) 共同完成. 它们之间必须协调动作互相配合, 甚至需要交换信息进行通信, 如: 任务和其他任务及 ISRs 交换数据, 任务与其他任务同步. 因此必须保证任务能对共享资源进行互斥的访问^[1].

为了满足任务间通信同步和互斥的需要, 同时保证资源被安全的使用, 必须对多个相关任务在执行的次序上进行协调. 本文设计的系统在消息同步通信机能上主要提供 Event Flag、Semaphore、Mail box 三种方式. 一个基本的嵌入式操作系统消息机制基本包括: Task 调度、Task 管理、同步通信、中断管理、时间管理机能等^[2,3].

1 TASK 调度

从整体上说, Task 调度机能就是 os 在对任务操作时所采用的基本方针. 从其功能内容来讲, Task 调度机能管理并决定任务的执行顺序, 并且赋予任务 CPU 的使用权. 当有如下几个条件发生时, 会发生任务的调度^[4]. (1)任务执行完毕; (2)发生改变系统运行状态的系统调用; (3)Dispatch 程序运行并对任务进行调度.

2 TASK 管理

系统中把任务分如下 5 种状态进行管理. 在任意时刻, 任务状态只能属于这 5 个状态之一. 在系统的任务管理和调度中, 任务在这 5 种状态之间切换. 这五种状态分别是: 未登录状态、就绪状态、运行状态、等待状态、睡眠状态. 图 1 表示了本系统的状态迁移.

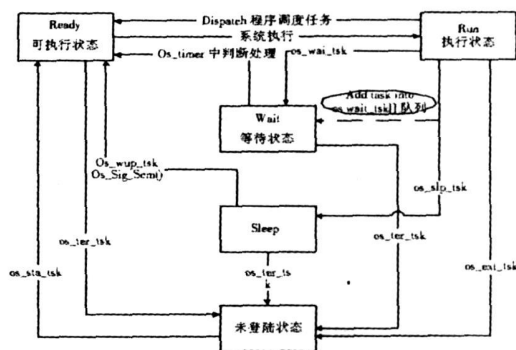


图 1 系统的状态迁移

3 同步通信

在多任务系统中, 存在资源的共享与进程的合作, 为了安全的使用资源, 必须对多个相关任务在执行的次序上进行协调, 所用的方法就是提供

* 作者简介: 高睿(1975-), 男, 辽宁沈阳人, 讲师, 硕士, 从事嵌入式应用研究.

收稿日期: 2007-03-22

Event Flag (任务间传递 Flag 信息), Semaphore (对系统资源进行排他访问), MailBox (任务之间进行信息通信) 机制。

3.1 EventFlag

在多任务处理系统中, 有时需要等到一个任务终了后, 其他任务才开始启动, 这个时候, 需要拥有对其他任务是否终了进行判断的能力, 而 EventFlag 提供了这种判断能力。

本系统中 Event Flag 所提供的功能是消息传递, 传递的是一个 32 位的 Flag 信息. 通过两个基本调用 msg_VSendEvent; 向任务发送 Event ; msg_VManageEventFlag; 在 Task 函数中调用, 用来导向处理 Event. 如 Task A 需要向 Task B 传递消息, 但是 Task B 处理需要的信息量很小, 一个 32 位的 Flag 就可以完全囊括, 这样系统会选择使用 Event Flag 来传递这个消息。

3.2 Semaphore

Semaphore 是系统对资源进行管理的一种机制. 系统中, 往往存在着多个 task 资源共享的情况, 但有些资源一次只能为一个任务使用, 为防止出现多个任务同时对资源的抢占, 本系统应用了 semaphore 对资源进行控制和管理. 为了实现 Semaphore 机制, 我们定义了如下函数来实现: (1) OS_VsetSemReq; Sets semaphore request (2) OS_VclrSemReq; Cancels semaphore request (3) OS_BchkReqSemSub; Checks semaphore request (4) OS_SigSem; Release a semaphore (5) OS_WaiSem; Waits semaphore (6) OS_GetSem; Gets semaphore information, 其它细节这里不在详述。

3.3 MailBox

一个邮箱是应用在其内存定义的一个长字变量. 允许任务传递 32 位的非 0 消息. 系统不创建邮箱, 而是由应用在其内存中建立邮箱. 应用应将邮箱初始化为一个恰当的值; 为 0 时, 邮箱可用, 当邮箱用于互斥时为非 0。

3.3.1 实现机制

系统的消息传递机制是整个系统成功运行的关键, 它是为了在任务间传递数据而设置的通信机制, 任务可以向邮箱写入数据或从中读出数据. 它的功能实现如下, 模块通信时传递的消息是一

个预先定义的数据结构 MSG-BOX-S, 每个模块自己维护一个这样类型的变量即邮箱并提供一个外部函数供其它模块操作本模块的邮箱, 这个函数的功能是通过邮箱变量调用系统函数从而把信息处理导向系统. 例如, 模块 A 要向模块 B 发送消息的话, A 会调用 B 提供的外部函数来实现对 B 邮箱的操作。

我们采用了 FIFO 方式的等待机制, 如图 2。

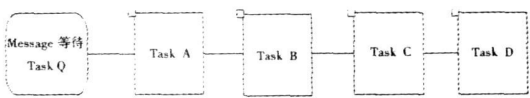


图 2 FIFO 等待机制示意

3.3.2 对邮箱的基本操作

①创建邮箱: 系统提供两种方式创建邮箱, 静态注册和动态创建. 系统动态创建邮箱是通过 cre-mbx 调用来实现的. ②删除邮箱: 系统提供的 del-mbx 功能可以实现邮箱的删除, del-mbx 发行后, 如果等待队列中存在任务的时候, 这些任务的状态将自动迁移到 Ready 状态. ③发送 Event: 系统提供的 snd-msg 调用实现 Event 的发送, 当 snd-msg 发送之后, 处于等待 Event 的任务的状态将由 Wait 状态迁移到 Ready 状态. ④接收 Event: 系统提供三种方式来实现 Event 的接收, rcv-msg, prcv-msg, 和 trcv-ms 一般来讲, 邮箱是任务之间信息通信过程中的场所, 本系统的 MailBox 中消息的暂时保存和传递不是由 OS 来管理而是由 Task 及所在模块自己来管理的. 在本系统中不存在真正 Mail-Box 需要的多任务应用的等待队列和邮箱专用的消息队列, 它的消息队列是作为静态量定义在每一个需要收发消息的模块中的。

使用 Mail Box 实现 Task 之间通信的处理流程如图 3 所示。

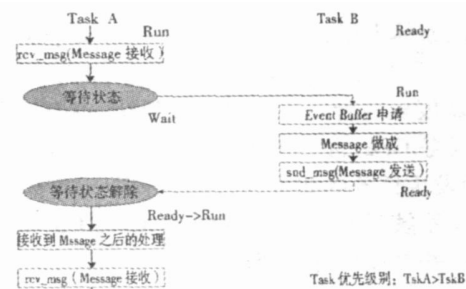


图 3 使用 Mail Box 实现 Task 之间通信的处理流程

在本系统中消息分发和处理的实现很简单,举个例子 Task A 向 Task B 发送消息的话,Task A 将调用模块提供的外部函数,这个函数会通过模块 A 的邮箱变量将处理权转移给系统提供的函数 msg_LSendMessage,后者将这个信息写入 Task B 的邮箱中然后调用系统函数创建一个 B Task。如果 B Task 被执行, B 所做的第一件事就是检查自己邮箱中是否有未处理的消息,然后进行处理。

消息流程如图 4 所示。

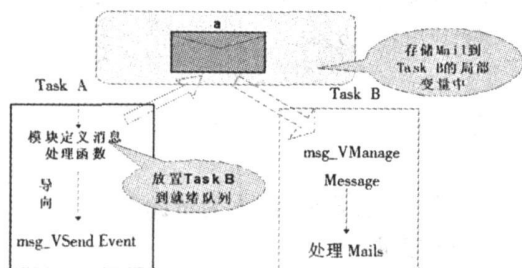


图 4 Task 利用 Mail Box 进行通信

4 中断处理的管理及时钟管理

中断分成两个处理步骤来进行中断处理。(1)中断 Handler,可以在短时间内完成,它并不真正执行中断处理,而仅是判断出中断类型,然后转到相应的中断处理函数去处理。(2)中断处理函数。在这里真正完成相应中断所要完成的各种

功能。

在实时系统中,时间是操作任务过程中最基础的要素,在内核中的周期唤醒、延迟唤醒、Time Out 等的操作都以它来作为基础。

5 结论

好的消息传递机制是一个系统成功运行的关键,而它实现的方式也多种多样。简单的说,本系统的消息传递机制是一个概念、两种方法,所谓两种方法,就是整个系统有两种消息通信方法,而一个概念就是说这两种实现方式的本质是一致的,只是用来处理不同的模块时,使用的是不同的方法而已。本方法在基于 ARM 内核的 S3C44BOX 系统中测试完全能达到性能要求。

参考文献:

- [1] 桑楠. 嵌入式系统原理及应用技术[M]. 北京: 航空航天大学出版社, 2002.
- [2] 田捷. 嵌入式软件开发[M]. 北京: 国防工业出版社, 2003.
- [3] 汤子瀛. 计算机操作系统[M]. 西安: 西安电子科技大学出版社, 2001.
- [4] Tae Rim Park, Jae Hyun Park, Wook Hyun Kwon. Reducing OS Overhead For Real Time Industrial Controllers With Adjustable Timer Resolution[M]. ISIE, 2001, 369—374.

The Research on the Message Mechanism in Embedded Operating System

GAO Rui, ZHANG Ming-yue

(The Department of Information Engineering, Liaoning Radio and TV University, Shenyang 110035, China)

Abstract: The realization of message mechanism in minitype embedded operating system is researched. It is divided into five parts for research; Task dispatch, Task management, synchronous communication, interrupt management, time management function. Aiming to ensure the correspond of various relative tasks in execute sequence, we elaborated with emphasis on the Event Flag, Semaphore, MailBox used in the synchronous communication mechanism, and stressing on the realization of Mail Box in synchronous communication. After running, it is proved that the system fully satisfies the need of communication.

Key words: embedded system; News mechanism; TASK; mailbox.

(责任编辑 郑绥乾)