

嵌入式操作系统 RTEMS 内核的初步分析

北京航空航天大学 王祖林 丁晓宇
北京控制研究所 华更新

摘要 在分析 RTEMS 内核的基础上,介绍嵌入式系统 RTEMS 内核的基本概念、基本组成结构、关键的数据结构、实时设计思想。文中还提出 RTEMS 对一些关键问题的处理方法。

关键词 嵌入式操作系统 双链表 优先级反转

一、RTEMS 简介

RTEMS(The Real Time Executive for Multi-processor Systems)为多处理器实时嵌入式操作系统,由美国陆军导弹指挥部于 1988 年研发完成。RTEMS 使用 C 和 Ada 两种编程语言,支持大多数的处理器芯片。目前,On-line Applications Research 公司提供 RTEMS 的后续开发和技术支持。可以免费获得 RTEMS 源代码,便于研究和开发。

二、RTEMS 内核的概述

RTEMS 内核操作是围绕任务进行的,没有 Windows 进程和线程的概念。在 RTEMS 中任务和线程是同一个概念。任务在计算机文献中有多种定义,然而,没有一种定义是与其操作系统无关的。RTEMS 将任务定义为竞争资源的最小执行单位。任务在系统中有四个状态:静止态、就绪态、执行态、阻塞态。其关系如图 1 所示:任务创建后处于静止态,并不参与任务调度。只有当任务启动进入就绪态时,任务才参与调度。根据调度算法,使某个任务占用处理器从而进入执行态。在就绪态或执行态的

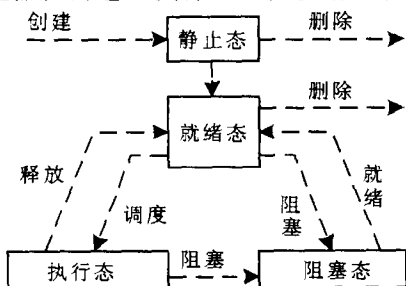


图 1 任务状态图

任务因为某些原因阻塞时,进入阻塞态。在静止态或就绪态时可将任务删除。

RTEMS 支持多任务,因此引入了任务队列的概念;多个任务之间要同步和抢占系统资源,因此引入了互斥量、信号量的概念;为了任务间的通信引入了消息队列概念;任务调度算法引入了优先级概念;同时为了使任务和时间联系起来,引入了看门狗的概念。这样就将整个内核围绕任务联系起来。

三、RTEMS 内核的分析

1. 对象管理

在 RTEMS 中,任务、消息队列、互斥量等都是对象,因此,RTEMS 提供接口函数动态的创建、删除、操作预定义的对象类型。所有的对象由应用程序创建并由 RTEMS 分配对象 ID 和用户分配对象名称。在内核操作中区分操作对象使用的是对象 ID,可以多个对象使用同样的对象名称。对象 ID 是唯一的无符号 32 位单元。其结构如图 2 所示。图 2 中的结点是对象创建的处理器结点,因此,同一种对象类型最多有 65535 个对象。

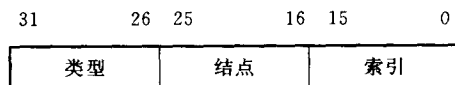


图 2 对象 ID

为了提高实时性,RTEMS 对象的管理采用了双链表结构。为此,RTEMS 提供了链表结点 `chain_node` 和链表控制 `chain_control` 两种数据结构。RTEMS 双链表和普通的双链表不同,它既可以看作是有头尾结点的双向链表,也可以看作是有头结点的双向循环链表。同时,此双向链表没有数据域。

没有数据域使双向链表具有通用性和灵活性,对于不同的对象类型只需将链表结点与对象数据域构成新的结点即可,而对于链表插入、删除等操作具有一般性。如对象结点 struct { Chain_Node Node; Objects_Id id; Objects_Name name; } Objects_Control,其中对象 ID 和对象名称就是所要加入的数据域。双向链表的结构如图 3 所示。

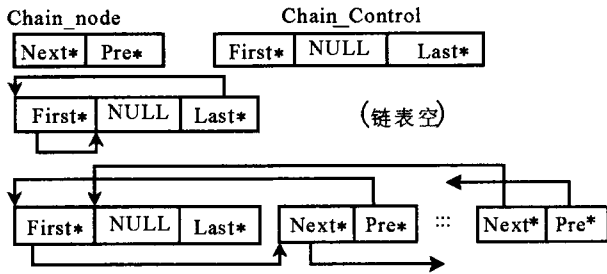


图 3 双向链表

2. 优先级管理

RTEMS 是基于优先级的抢占式调度。RTEMS 支持 256 级优先级(0~255),数字越小优先级越高,0 的优先级最高。RTEMS 使用 _Priority_Major_bit_map 和 _Priority_Bit_map[16] 两个数据结构来控制完成任务调度。如图 4 所示,每个数据单元都是 16bit。_Priority_Major_bit_map 的每一位代表 16 个优先级中是否有就绪任务。_Priority_Bit_map[16] 中的每一位代表某个优先级是否有就绪任务。置位代表存在就绪任务。通过这两个数据结构就可以判断当前应该调度的任务的优先级。如图 4 所示:当前应调度的就绪任务的优先级数为 $3 \times 16 + 1 = 49$ 。搜索方法从左至右,直到找到有就绪任务为止。

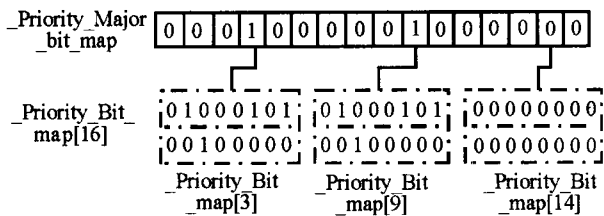


图 4 优先级数据结构

3. 任务的管理

为了管理每个任务,提供了任务控制块(TCB)数据结构。RTEMS 为每个任务分配其唯一的 TCB 和任务堆栈。任务控制块中有几个重要的组成部

分:(1) 任务的等待信息。当任务处于阻塞态时,等待信息包含阻塞等待的对象 ID,任务所处的阻塞任务队列。(2) 任务的上下文控制,提供任务切换需要保存的内容。(3) 任务就绪链表。当任务进入(脱离)就绪态时将任务向(从)链表中添加(删除)。

为了实现任务之间的同步和互斥,RTEMS 提供了互斥量和信号量。互斥量用于单个任务的同步和任务对独占资源的访问;信号量用于多个任务的同步和任务对非独占资源的访问。对用户而言,操作系统的互斥量和信号量的操作简单,提供的接口就是对互斥量和信号量的创建、获取和释放操作。

为了实现任务之间的通信,RTEMS 提供消息队列。如图 5 所示,任务 A 将消息放入消息队列,任务 B 从消息队列中取消息。消息队列为内核分配一定大小的缓冲区。通常,消息队列符合 FIFO 原则;紧急情况下,符合 LIFO 原则。因为任务之间的通信是通过消息内存的搬移来完成的,所以,放在消息队列中的数据量不应过大,否则,会影响通信的速度。若需要通信的数据量大时,可考虑向消息队列中传递数据指针或开辟共享内存进行通信。

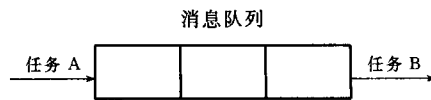


图 5 消息队列结构

RTEMS 提供了看门狗。如任务等待一段时间启动,任务阻塞一段时间后重新成为就绪态等都需要用到看门狗。看门狗的刷新是与系统的外部时钟相关的,外部时钟的频率不能过高(一般考虑 60Hz)。因为每次外部时钟触发的中断都会刷新整个系统的状态,因此,会耗费一定的系统时间。若频率过高,就使得用户的程序没有时间运行。

任务有阻塞态,这就产生了优先级反转现象。如图 6 所示,首先是低优先级的任务在执行并获得互斥体进入临界区。当中等优先级的任务就绪时,

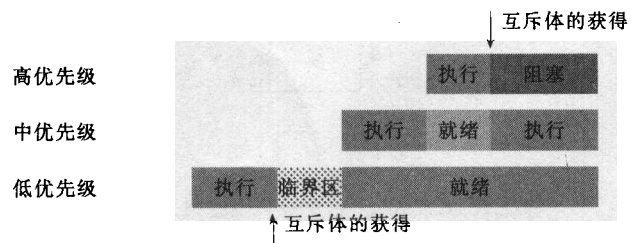


图 6 优先级反转现象

抢占低优先级的任务进入执行态,同时低优先级的任务进入就绪态;当高优先级的任务就绪时,又抢占中优先级的任务进入执行态,同时中优先级的任务进入就绪态。当高优先级的任务也要获取互斥量,进入阻塞态时,中优先级的任务又从就绪态进入执行态。高优先级的任务只有当中优先级的任务执行完,同时低优先级的任务释放互斥量时,高优先级的任务才能执行。这显然和高优先级的任务最先执行相矛盾。为了解决这个问题,RTEMS在任务控制块中提供两个优先级:一个是实际优先级,一个是当前优先级。如图6,当高优先级的任务要求获取互斥量时,低优先级的任务的当前优先级提高为高优先级任务的优先级。低优先级的任务释放互斥量后恢复低优先级任务的当前优先级。这样就会使高优先级的任务阻塞后低优先级的任务先执行,释放信号量后使高优先级的任务迅速开始执行。

4. 中断管理

RTEMS支持256级中断。如图7所示,外部

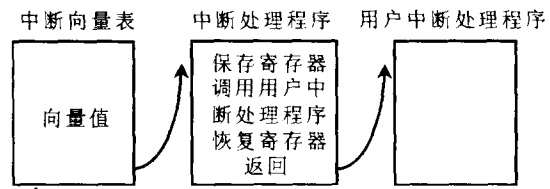


图7 中断响应流程

中断产生后,首先根据中断向量值查找中断向量表,调用相应的中断处理程序。在中断处理程序中保存被中断任务的内容并调用用户中断处理程序。恢复被中断任务的内容,中断返回。从中断产生到用户中断处理程序处理完毕,就是系统相应中断响应的时

参考文献

- 1 On-line Applications Research 公司. RTEMS 源代码文档
- 2 华力同创公司. Vxworks 培训教程
- 3 唐发根,等. 数据结构教程. 北京:北京航空航天大学出版社,1996
- 4 姜文清. 编译技术原理. 北京:国防工业出版社,1994

(上接第32页)

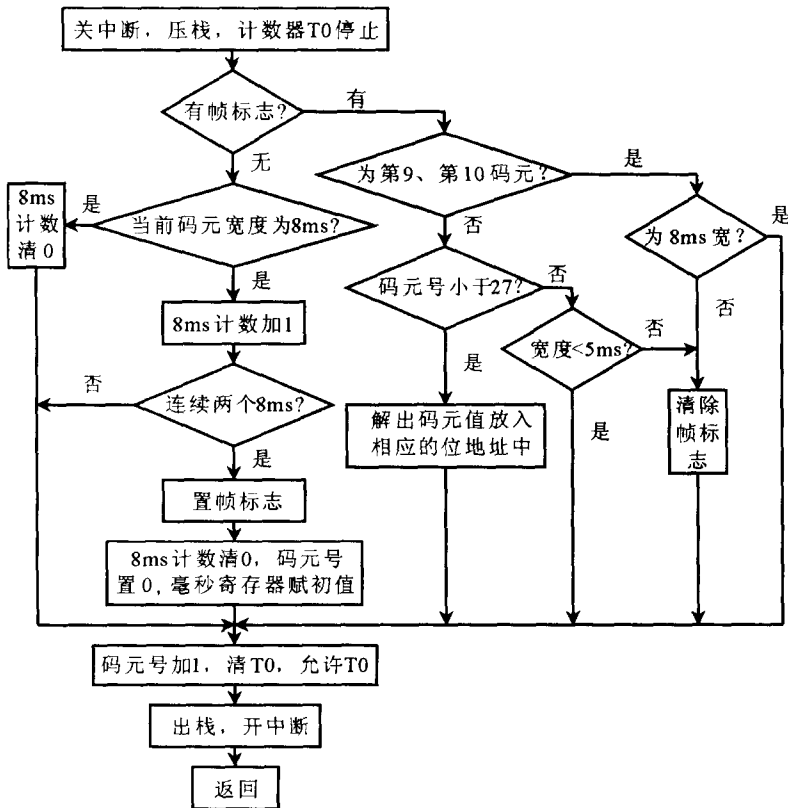


图5 INT0中断处理程序流程图

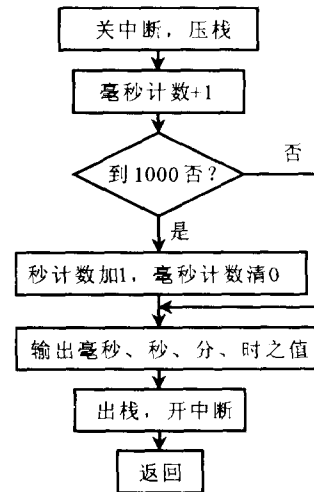


图6 INT1中断处理程序流程图

参考文献

- 1 何立民. MCS-51 系列单片机应用系统设计. 北京:北京航空航天大学出版社,1990
- 2 李华. MCS-51 单片机实用接口技术. 北京:北京航空航天大学出版社,1993