第35卷 第13期 计 算 机 工 程 2009年7月
Vol.35 No.13 Computer Engineering July 2009
·工程应用技术与实现· 文章编号：1000—3428(2009)13—0249—03 文献标识码：A 中图分类号：TP303

基于 VxWorks 的多中断处理设计

唐晓平，何 峰，梁甸农
(国防科学技术大学电子科学与工程学院，长沙 410073)

摘 要：考虑 VxWorks 的中断在具体实现上的特殊性，提出一个针对多块采集板系统的中断设计方案，把传统的中断服务程序分解为中断服务程序和中断服务任务，利用信号灯进行同步，通过在中断服务任务中引入用户定义的优先级，增强系统处理多中断时的可靠性和实时性。

关键词：实时操作系统；多中断；中断服务程序；中断服务任务；优先级

Design of Multiple Interrupts Treatment Based on VxWorks

TANG Xiao-ping, HE Feng, LIANG Dian-nong
(School of Electronic Science and Engineering, National University of Defence Technology, Changsha 410073)

【Abstract】Considering the specificity in the specific realization of interrupt in the VxWorks, this paper proposes an interrupt treatment design of a system consisted of multiple data acquisition boards. It divides the traditional Interrupt Service Routine(ISR) into interrupt service routine and interrupt service task, and makes use of the semaphore to synchronize them. By introducing the priority that users define, the design enhances the real-time performance and stability of the system.

【Key words】RTOS; multiple interrupts; Interrupt Service Routine(ISR); interrupt service task; priority

VxWorks 是目前主流的商用实时操作系统，它为程序员提供了高效的实时任务调度、中断管理、实时系统资源管理以及实时的任务间通信。如果在某半实物仿真系统中开发一个由多块采集板构成的数据采集系统，并使用 VxWorks 作为整个系统的主控软件，那么在整个系统的控制中，多中断设计成为一个极为重要的问题。本文针对项目的实际情况对多中断处理展开讨论，提供了一种针对多数据采集板系统进行中断管理的方法。

1 VxWorks的中断处理机制

硬件中断处理是实时系统设计最重要、最关键的问题。中断通常对应外部事件，系统通过中断与外部事件交互。为了获得尽可能快的中断响应时间，VxWorks 的中断处理程序运行在特定的上下文中，它在所有任务上下文之外。因此，中断处理不会涉及任何任务上下文的交换。VxWorks 的库 intLib 和 intArchLib 提供了中断处理的相关函数。对于使用了内存管理单元的底板，VxWorks 还提供了可选产品 VxVMI 以提供中断向量写保护机制。通常情况下中断处理的过程如图 1 所示[1]。

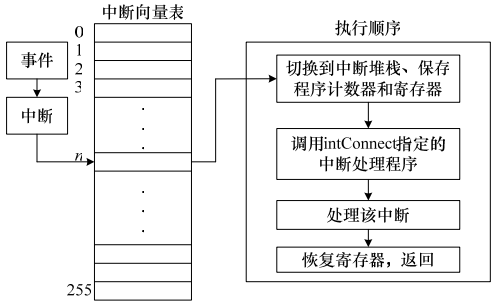

图 1 中断处理过程

应用程序可以使用中断向量表中未使用的中断号资源。VxWorks 提供函数 intConnect()，它允许将指定的 C 函数与任意中断相联系。intConnect()函数原型如下：

```
STATUS intConnect{
VOIDFUNCPTR * vector; /*要联系的中断向量*/
VOIDFUNCPTR  routine;/*中断发生时要调用的函数*/
int      parameter;  /*传递给中断处理函数的参数*/}
```

该函数将指定的 C 函数 routine 与指定的中断向量 vector 相联系，函数的地址将存储在这个中断向量里。所以，这个中断发生时，系统将调用该函数，使用指定的参数 parameter 作为参数。中断处理程序在中断级以 supervisor 方式调用，并建立一个合适的 C 环境，保存必要的寄存器，建立堆栈。中断处理函数可以是任何正常的 C 代码。但是它必须保证不调用任何可能引起阻塞和执行 I/O 操作的函数。该函数简单地调用 intHandlerCreate()和 intVecSet()。处理程序的地址由 intHandlerCreate()函数返回，保存在中断向量中。intConnect()将创建一小段代码，用以保存必要的寄存器，设置堆栈入口，包含将要传递的参数，在堆栈中调用这个连接函数。相反，当从该函数返回时，这段代码先恢复寄存器和堆栈，然后退出中断，如图 2 所示[2]。

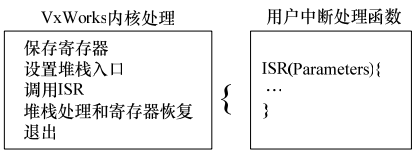

图 2 用 intConnect()构造的用户中断处理函数

基金项目：国家部委基金资助项目
作者简介：唐晓平(1984－)，男，硕士研究生，主研方向：嵌入式系统；何 峰，副研究员；梁甸农，教授、博士生导师
收稿日期：2008-11-20 **E-mail：**txp914111@126.com

2 传统多中断处理存在的问题

由于用函数 intConnect()调用的用户中断处理函数在中断级以 supervisor 方式调用，因此为了保证 VxWorks 的实时性，应该尽可能在最短的时间内完成操作，以免影响实时系统对高频突发事件的响应。另外，VxWorks 的 ISR 必须遵循一个基本约定：不能调用可能引起阻塞的函数和 I/O 操作函数。如 ISR 不能调用 semTake()来获取一个信号量，因为在此调用中，一旦发现信号量不可用，调用者会随即转入阻塞状态。此外，诸如 malloc()和 free()等函数在实现时必须获取一个信号量，所以，也不能被 ISR 调用。而且 ISR 不能调用那些使用浮点协处理器的函数。因为在由 intConnect()创建的中断驱动代码中不保存和恢复浮点寄存器。另外，在多中断系统中，如果一些中断处理函数长时间占用中断号，容易出现系统中断号资源不够用的情况，可能导致中断请求超时而引起系统出现不可预测的后果，不能保证 VxWorks 的可靠性。

3 基于VxWorks的多采集板系统中断处理设计

在多采集板的大型系统中，通常采用工业机箱为多块板卡统一提供电源、时钟和仲裁信号，并使用零槽的主控板对各采集板进行统一管理，使各板卡按照一定的时序协同工作[3]。各采集板在数据采集完成后会向系统发出中断请求，然后进行数据的传输。因此，按照传统中断管理方法，数据传输是在 ISR 中完成的。但是，由于实际的数据量较大，而传输带宽有限，因此会导致 ISR 执行的时间过长，严重影响系统的强实时性和可靠性。针对这个问题，下面介绍一种应用实时操作系统 VxWorks 进行多中断处理的方法。

3.1 多中断处理的程序设计框架

为了能尽快响应中断，VxWorks 中的中断服务程序不在固定的任务上下文中执行，而且没有任务控制块，所有中断程序必须共享一个单独的堆栈，这导致许多 VxWorks 函数在中断服务程序中被禁止使用。由于中断程序中不能调用可能导致阻塞的任务，因此中断服务程序的函数本身受到很大的局限，对于必须紧接着中断而进行的网络或者 I/O 等操作，可以通过拆分中断服务程序来完成，即将原来的中断服务程序拆分为中断服务程序和中断服务任务 2 个部分：新的中断服务程序仅仅执行最基本的中断处理，例如禁止中断、判断中断类型；绝大多数的任务处理，特别是会造成阻塞的任务应该在中断服务任务中执行。中断服务程序和中断服务任务使用信号灯同步，其程序框架如下：

```
void intFunc(int param)     /*中断服务程序*/
{intDisable(intNum);        /*禁止中断*/
…
semGive(semInt);            /*释放同步信号灯*/
intEnable(intNum);          /*重新允许中断*/}
void intHandle()            /*中断服务任务*/
{while(semTake(semInt,WAIT_FOREVER)!=ERROR/*若获取信号灯成功，则执行相关工作*/
…}
```

3.2 多中断的硬件控制简介

本系统中采用了 CompactPCI 机箱 SCx784T，它是 IEEE1101.10 兼容的 8 槽背板机箱，其中，零槽被固定为系统槽，处理整个 CPCI 机箱，包括对各插板进行配置、中断处理、热插拔控制等。在零槽中使用 CK5 作为主控单板机，它使用 PowerPC 系统控制器 MV64460 对系统提供了几个集成功能，如 IDMA 控制器、中断控制器，它对中断控制的电路如图 3 所示。CK5 使用 CPLD 寄存器获取外部(cPCI 和 PMC)中断和板内中断。当其中一个中断被声明后，CPLD 就向 MV64460 的多目的管脚(MPP)声明一个中断。MV64460 获得中断请求后就在中断控制器中设置引起中断的寄存器的管脚。这样，中断控制器就向处理器 MCP7447A 声明了中断 CPU_INT0[4]。

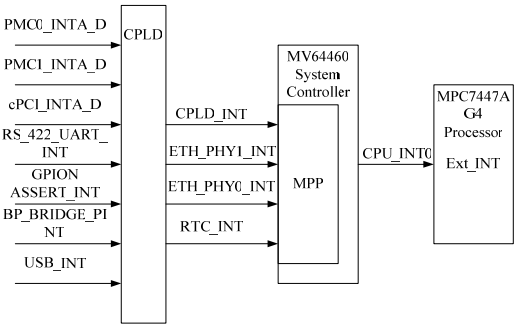

图 3 主控单板机 CK5 中断的硬件控制结构

3.3 多中断处理的初始化

多中断处理的初始化主要是对 cPCI 机箱中各采集板指定的 PCI 首部寄存器进行读写操作，完成全局变量的初始化、用户中断函数的绑定和多个任务的发起。首先调用系统函数 pciFindDevice()查找各插槽中的采集板，这个函数根据各板卡上 PCI 桥的设备标识和厂商标识以及板卡所在的槽号识别采集板。识别出采集板后，调用系统函数 pciConfigInLong 读取基地址寄存器，并根据寄存器中的内容对采集板分配所需的存储器空间和 I/O 空间，将分配的空间写入对应的基地址寄存器。根据 PCI 中断请求引脚和中断控制器的输入关系，调用 pciConfigOutByte 写 0x3C 处的中断线寄存器；读取 0x3D 处中断引脚寄存器以确定设备(或功能)使用了 INTA#~INTD# 中的哪个引脚。

根据 PCI 规范，所有 PCI 设备都将复用 4 个中断级别，即 INTA#~INTD#。超过 4 个 PCI 设备的多采集板系统不可避免地存在中断复用的问题。要实现不同设备复用同一中断，不但要在硬件上使用电平触发机制，还要在软件上做相应处理。在这种情况下，如果多个中断服务程序使用相同的中断向量，那么后接入的中断服务程序将会覆盖前面的中断服务程序，等挂接多个中断服务程序到相同的中断号后，只有最后一个中断服务程序是起作用的，无法达到中断复用的目的。因此，VxWorks 在系统内部对 PCI 的 4 个中断源创建了 4 个保存复用中断服务程序入口的中断服务链表，并使用 pciIntConnect()函数把中断服务程序挂接在对应的中断服务链表中。在某个中断号对应的中断引脚出现低电平时，VxWorks 会使用函数 pciInt()逐个执行对应中断号的中断服务链表上的中断服务程序，由中断服务程序检查与设备中断相关的寄存器，判断是哪一个设备产生的中断。

3.4 多中断处理的实现

多采集板的中断处理由中断服务程序和中断服务任务协同完成，其处理流程如图 4 所示。在硬件中断发生后，先需要保存当前系统寄存器状态以及正在执行的代码，然后装载中断堆栈。在完成这些获得中断后所需的最基本的工作后，CPU 根据引起中断的寄存器管脚判定出中断号，调用连接在与之对应的中断向量上的中断服务程序 usrIntFunc(intNum)。中断服务程序的首要工作是：通过查询所有挂接到此中断的插板的指定寄存器，判断其指定的寄存器是否等于特定值，由此确定此中断由哪块插板产生。然后释放相应的信号灯，

并使能该中断号。中断服务任务 usrIntHandle(slotNum)循环获取信号灯，在没有获得由中断服务程序释放的信号灯之前，任务处于阻塞状态，这时任务不占用 CPU。如果成功获得信号灯，此任务将中断计数器加 1，再进行相应的处理，如数据的处理和存储，完成了用户设定的任务之后允许此插板再次传输数据，这样此板就可以产生下一次中断。中断任务在运行期间不像 ISR 那样以 supervisor 级别运行，用户可以根据不同中断的紧急情况为不同中断任务设置优先级别，以保证最紧急的中断任务优先执行，从而保证了系统的可靠性。

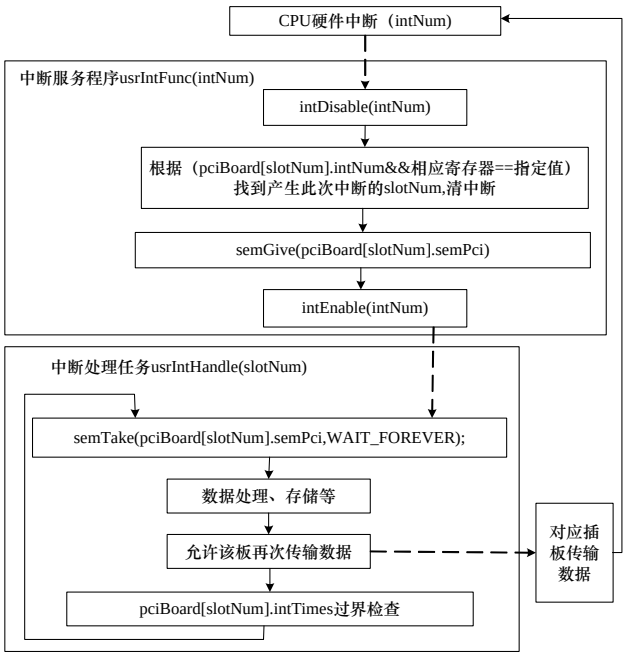

图 4　多中断处理流程

在 VxWorks 中，信号灯是提供任务间通信、同步和互斥的最优选择，它提供任务间的最快通信，也是提供任务间同步和互斥的主要手段[5]。这里使用的计数型信号灯还可以提供缓冲功能，当一个中断任务尚未执行完毕前，可以由计数型信号灯保存在这一时期发生的中断请求信息[6]，使别的中断不会因为中断请求超时而丢失。另一方面，由于这个中断号已经在中断服务程序中使能了，因此在这个插板的中断处理完成之前，中断服务程序仍然可以响应挂接在该中断号上的另一块插板所产生的中断。这样就可以更有效地利用系统有限的中断号资源，并加快系统响应多中断的速度，提高系统的实时性。

4　结束语

由中断服务程序和中断服务任务利用 VxWorks 的信号灯协同完成多采集板系统的中断处理方法充分利用了 VxWorks 对多任务的良好支持，可以在中断处理任务阶段实现基于优先级的中断处理，从而保证系统的稳定性；同时可以及时释放中断号，加快中断响应的速度，发挥 VxWorks 的强实时性。另一方面，使用中断处理任务还可以弥补 ISR 由于不能调用某些函数而受到的限制，增强了中断处理的能力。

参考文献

[1] 孔祥营，柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境[M]. 北京：中国电力出版社, 2002.

[2] 邝　坚. Tornado/VxWorks 入门与提高[M]. 北京：科学出版社, 2004.

[3] 程敬原. VxWorks 软件开发项目实例完全解析[M]. 北京：中国电力出版社, 2004.

[4] SBS Technologies Inc.. CK5 User's Guide[EB/OL]. [2008-06-13]. http://www.sbs.com.

[5] Wind River Inc.. VxWorks Programmer's Guide 5.5[EB/OL]. [2008-06-23]. http://www.windriver.com.

[6] 张　皓，伍　云，周志杰. 基于 VxWorks 的多任务间通信模型设计[J]. 计算机工程, 2007, 33(3): 131-132.

编辑　张　帆

（上接第 248 页）

控制 CAN 设备工作的功能软件，提供基础类库的各种函数集，以及实现新一代加油设备核心控制系统智能加油任务的模块化功能进程集。所有功能相对独立的任务都基于 ICS 通信机制被进程化为独立软件模块来实现，最大化实现了软件模块内部高内聚、模块间的低耦合，增强了控制系统的扩展性和开放性。

4　结束语

本文提出的新一代电脑加油设备控制系统的设计思路已经赋之于实践，目前已经制作完样机，开发管理和子模块测试效果很好。它充分利用了 32 位 ARM9 微处理器和嵌入式 Linux 的优势，在此宽适应性的基础运算与管理平台上，基于 CAN 总线来设计相对独立的硬件功能板；基于 ICS 机制架构进程化软件功能模块，使软硬件模块都最大化相对独立且性能稳定、使用灵活。同时借助于新一代控制系统丰富的系统资源和操作系统强大的设备管理能力，单芯核心板实现 4 枪同时加油功能，取代 4 芯多控制板仅能实现单枪加油的现状。这些均赋予电脑加油设备新的技术特征，在降低设备成本及提高开放性、智能性和可维护性方面具有显著优点，值得在行业内推广。

参考文献

[1] 邢远秀，于继武，方　明，等. 嵌入式加油机刷卡系统的研究与开发[J]. 武汉理工大学学报：交通与科学工程版，2006，30(5): 923-926.

[2] Guo Bing, Shen Yan. Hardware-Software Partitioning of Embedded Operating System in the SoC Using a Discrete Hopfield Neural Network Approach[J]. Chinese Journal of Electronics, 2007, 16(1): 13-17.

[3] 万相奎，徐　杜，张　军. 基于嵌入式系统的科学仪器的构建[J]. 计算机工程, 2007, 33(14): 213-215.

[4] Yang Wu, Fang Binxing. Research and Improvement on Packet Capture Mechanism in Linux for High-speed Network[J]. Journal of Harbin Institute of Technology, 2005, 12(5): 494-498.

[5] 孙　兵，何　瑾，陈广厦. 基于 DSP 的 CAN 总线与以太网互联系统研制[J]. 仪器仪表学报, 2008, 29(2): 377-380.

编辑　顾逸斐