

文章编号:1001-9081(2005)02-0443-03

Windows 2000/XP 下通用实时采集的设计与实现

吴丽娜,高敬阳

(北京化工大学 信息科学与技术学院,北京 100029)

(wlenna@126.com)

摘 要:首先分析了 Windows 2000/XP 平台上几种实时采集方案,然后综合利用 Windows 2000/XP 为实时应用所提供的机制,提出了一种基于非专用采集卡的高精度实时采集方案。最后通过某振动信号采集系统的实现,给出该方案的详细设计思路和实验数据,证明了其正确性与可行性。

关键词:实时数据采集;Windows 2000/XP 操作系统;线程优先级;定时器;中断

中图分类号: TP311.1;TP316.7 **文献标识码:** A

Design and implementation of general real-time data sampling in Windows 2000/XP

WU Li-na, GAO Jing-yang

(School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: The paper analyzed some methods of real-time data sampling in Windows 2000/XP platform and proposed an optimal solution using the mechanisms for real-time application in Windows 2000/XP. For disserting the design idea, A vibration signal sampling system was proposed as an example application and experimental data was given which indicates that this solution can fulfill the requirements of real-time application. The practical result has proved to be successful.

Key words: real-time data sampling; Windows 2000/XP; priority of thread; timer; Interrupt

0 引言

实时数据采集是工业控制中一个很重要的环节,直接影响控制的质量,而当今 Windows 系列操作系统在 PC 机上占据着统治地位,它标准美观的图形用户接口(GUI)和方便快捷的操作方式,被普通用户所接受,一些实时系统领域的开发人员也在尝试把它用于实时数据采集系统。

1 实时采集的常用实现方案及分析

Windows 2000/XP 实现的是基于线程优先级的抢先式任务调度和中断调度,提供了 32 个线程优先级和 32 个中断优先级,同时还引入了延迟过程调用 DPC 机制和异步过程调用 APC 机制等等,目前常用的几种数据采集实现方案正是基于这些机制得以实现。

1.1 利用线程优先级和 Sleep 函数的多线程实现方案

在多线程的实现方案中,采集线程的优先级设为应用程序线程的最高级 15。采集线程不断对采集卡的状态进行查询,若数据准备好则立即将数据从卡上缓存取入 PC 内存,否则线程通过 Sleep 函数,暂时睡眠若干毫秒,将 CPU 资源让给显示、存盘、数据分析等低优先级线程。Windows 2000/XP 的抢先式调度机制能够保证采集线程苏醒后能立即抢先到 CPU。

但 Windows 2000/XP 只在时钟中断来临时,才执行线程的唤醒工作,所以这种方案的精度只能达到时钟中断的间隔

时间,通常为 10ms。

1.2 采用内核定时器

Windows 2000/XP 提供了两种定时器:用户定时器和内核定时器,用户定时器利用消息机制来激发系统的定时处理,而内核定时器则采用 APC 调用的方式来通知应用程序。但两者都是通过拦截时钟中断来实现计时,所以精度最高也只有 10ms。

1.3 多媒体定时器

多媒体定时器能按指定的精度设置计数芯片 8253 的初值,以获得任意精度的时钟中断,然后创建一个优先级为 15 的独立线程负责监测时钟中断,达到定时间隔时,采用 APC 的方式调用指定的处理例程。由于监测时钟中断的线程优先级只为应用程序的最高级 15,仍有可能被许多实时系统线程和中断抢先,因此它仍然具有一定的不确定性。

表 1 指定定时间隔下定时器触发 5000 次的最大间隔时间

实现方案	设定间隔(ms)			
	1	5	10	20
优先级	11.06	23.09	26.02	36.96
内核定时器	10.57	10.68	10.56	20.88
多媒体定时器	1.91	5.75	10.82	20.77

从表 1 中可以看出,采用 Sleep 函数进行定时带有很大的不确定性,内核定时器能够实现 10ms 以上的定时精度,而多媒体定时器,则能够达到毫秒级精度,其误差不超过 1ms。

收稿日期:2004-07-10

作者简介:吴丽娜(1977-),女,湖南怀化人,硕士研究生,主要研究方向:计算机控制;高敬阳(1966-),女,辽宁沈阳人,副教授,硕士,主要研究方向:计算机控制。

1.4 编写中断服务程序

轮询机制费时费资源,与轮询机制相比,中断处理方式要优越得多。Windows 2000/XP 采取了延迟过程调用 DPC 机制来保证对中断信号快的响应,通过实验发现,只要机器内所有硬件使用的都是采用 DPC 技术的驱动程序,Windows 2000/XP 的中断响应时间能均匀地分布在 $3\mu\text{s} \sim 7\mu\text{s}$ 之间,不发生中断丢失的现象,这样的响应速度对于精度要求在毫秒级上的实时应用来说,是完全可行的。

2 应用实例

某转子实验台平衡检测系统,要进行振动信号的整周期截取,通过快速傅立叶变换 FFT,取得与转速同频率的振动分量,最后利用影响系数法得出转子所需的平衡量。用 FFT 技术对信号进行频谱分析,栅栏效应和泄漏效应是主要的误差来源,栅栏效应是由周期数据的不完整造成,泄漏效应则是由时域截断产生。在采样频率足够高,且 FFT 处理的是原波形的完整周期的数据的情况下,这两类误差可以避免,因此,要保证动平衡的精度,必须实现振动信号的“整周期截取”。现场常用的方法是采用带有锁相倍频和并行采样/保持电路的动平衡专用采集卡,通过硬件实现整周期截取。专用采集卡的弊端在于它的不通用性以及难以维护和成本昂贵。由于本系统主要用于实验室教学,所以其设计目标是要在通用操作系统和通用采集卡上实现转子的动平衡。通用采集卡上没有用于整周期截取的锁相倍频电路,传统的软件异步倍频法又由于转子转速的波动而存在着很大的误差,为了达到与硬件相同的整周期精度,本系统采用了软件模拟锁相倍频的方法,即首先根据当前转速与倍频数换算出下一周期的采样间隔,然后在采样过程实时监测键相信号的上升沿,以确定整周期的开始与结束,这种方法以软件异频倍频为基础,模拟硬件同步倍频的原理,能够实现完整的整周期截取,其核心就在于对转速和键相信号上升沿的实时监测。同时,由于没有并行采样/保持电路,采集卡的工作方式为串行采集,要减小截取误差,做到在高转速下仍然保持所要求的倍频数,采样频率必须足够高。而采样频率的高低就决定了采集系统的实时响应时限。

针对以上的实时性要求,本系统主要从三个方面进行了构思。

2.1 采集卡的选择

系统选用阿尔泰公司的 12 位高速采集卡 PCI 2000。该卡带有 2kb 的 FIFO 缓存, A/D 转换时间为 $10\mu\text{s}$,缓存写满 1kb 时采集卡能发出缓存半满的信号。

这种乒乓双缓冲机制减轻了软件的压力,在一个缓冲区满的情况下发出中断请求软件处理,同时另一个缓冲区仍可继续接收采样数据。使得定时响应的精度从 $10\mu\text{s}$ 降低到了 5.12ms,为通用操作系统实现实时应用提供了可能。

2.2 中断服务程序的设计

中断服务程序将硬件与应用程序联系起来,因此它是实现系统实时性的关键,它不仅要处理采集卡的中断信号,在规定的时限内完成数据的接收,同时,它还要负责向应用程序提

供数据。

在本系统中,中断服务程序创建一个 64kb 的循环队列形式的一级缓冲。设备发出缓存半满信号时,操作系统响应中断,中断服务程序负责将数据取至一级缓冲区。一级缓冲属于系统内存空间,由中断服务程序负责维护,完全与应用程序独立,它的引入,实现了数据采集与数据处理完全独立的模式,使得应用程序的实时响应时限从 5.12ms 进一步下降到了 163ms。

中断服务程序完成数据转移后,为采集线程加入一个异步过程调用 APC 例程,然后发送核心信号通知该线程。

2.3 应用程序的设计

应用程序完成采集数据的一级缓存读入、存盘、波形绘制、整周期检测以及频谱分析等多项工作,如此繁重的任务,多线程是必须的。为了使数据得到及时接收处理,负责一级缓存的读入工作的采集线程设为实时优先级;要实现大量数据的不间断连续存盘,存盘线程也安排在一个较高的优先级上;要实时监测转速与键相信号,整周期检测线程的优先级必须高于一般线程;波形的绘制、频谱分析这一类工作,计算量大而费时,但实时性要求并不高,过于频繁的波形刷新,带来的将仅是闪烁的屏幕,而不是人眼能分辨的波形,因此,利用用户定时器,这类线程被安排为隔几百毫秒工作一次。

整周期检测线程是动平衡的核心,它实现了硬件锁相倍频的功能。即对所有采样数据进行监测,一旦发现上升沿,则表明一个周期结束和新周期的开始,线程将前一个周期的数据保存为一个数据块,通知各个处理线程来取整周期数据,然后根据实际的倍频数与采样频率计算出当前的转速,确定下一周期的采样间隔,最后开始新一轮的上升沿监测。通过这种方法,克服了软件异步倍频中由于转子转速的波动,而带来的异步误差,使得整周期截取的误差只受倍频数的影响,即采集卡可以达到的采样频率的影响,在本系统中,采样频率始终保持为采集卡的最高采样频率 100kHz。

在多线程环境中,为了保证实时不漏点采集和实现各个线程在速度上的匹配,本系统引入了三个缓冲区,即 BufferArray、BufferList、ShowList。

采集、存盘和整周期检测线程处理的是从一级缓存取回的数据,为它们建立的是大小为 32k 的循环静态链表 BufferArray,即二级缓存,链表节点为指针。由于整周期检测线程计算量大,耗时较多,为了不阻塞其他线程,为它专门建立一个队列 BufferList,存盘后的数据就从 BufferArray 移入该队列,等待整周期分析与计算。由于二级缓存的节点为指针,所以数据取入队列只需复制一个地址,而不是耗时的数据转移或拷贝。

波形绘制和各种频谱分析线程处理的是整周期数据,因此它们共用一个非循环队列,即显示队列 ShowList,节点为整周期数据指针。具体实现为:整周期检测线程将数据按周期组织成数据块,加入显示队列中,波形绘制和频谱分析线程从最新的整周期数据开始,对若干个周期数据进行指定处理。

对于多线程的同步与互斥问题,系统根据不同的线程合作方式,引用了多种信号机制。

中断服务程序与采集线程之间的通信是通过核心信号对象实现。采集线程通过 WaitForSingleObject 函数一直在不占用 CPU 资源地等待中断服务程序发来的信号,一旦检测到信号立即进行数据的二级缓存转移。采集线程的实时优先级能够保证它在得到信号后立即抢先到 CPU 资源。

采集、存盘和整周期分析线程之间的同步是利用全局变量实现,即全局变量 WriteIndex、SaveIndex 和 ShowIndex 分别记录它们在二级缓存中的操作位置,这三个变量的所有操作均为原子操作。采集线程每完成一次写入,WriteIndex 向后移,SaveIndex 指示存盘线程工作进度,它紧跟着 WriteIndex 向后移。所有已存盘节点均要移入整周期检测线程的工作队列中,由 ShowIndex 指出当前正在转移的数据。

整周期检测线程与波形绘制及其他频谱分析线程之间通过消息机制同步。整周期线程完成周期数据分析后,调用 PostMessage 向视图邮寄自定义消息 WM_TRANSOVER。视图收到消息后启动所需的各种频谱分析线程,并在主线程中完成波形绘制。

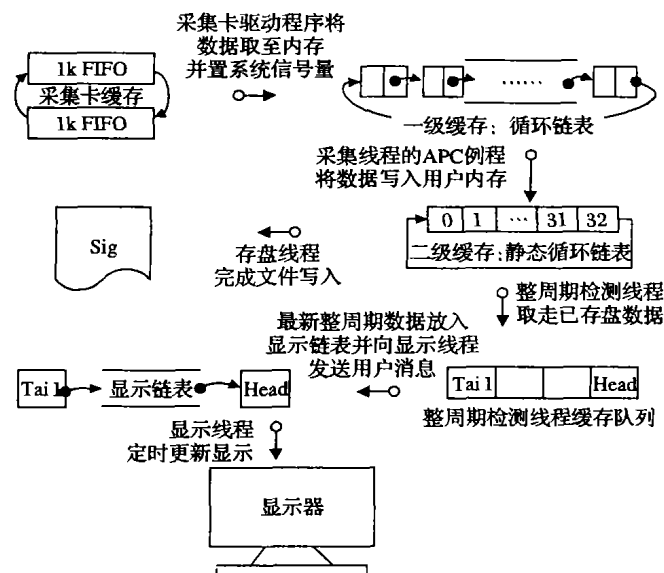


图1 转子平衡检测系统数据结构设计

3 误差与性能分析

系统采用软件锁相倍频进行整周期截取,因此其整周期截取的误差只受倍频数影响。系统始终工作在采集卡的最高采样频率 100kHz 上,在双平面平衡中,需要采集一路键相信号和两路振动信号,即使是在转子实验台的最高转速 200 转/秒(即周期为 5ms)的情况下,对每路振动信号也是进行的 170 倍频采样,即整周期截取的相位误差为 $360/170 = 2.1^\circ$,截取误差不超过 0.6%。整周期的截取误差将导致由 FFT 分析所得出幅值相位有一定误差,其中相位误差较大,通过计算可知其大小为整周期截取误差的 π 倍,即相位误差为 1.08° 。转子转速越低,各振动信号的每周周期倍频数将更大,整周期截取误差就将更小,因此在该平衡检测系统中,通过 FFT 分析所得出的相位误差不会超过 1.1° 。

振动信号通过采集卡的 A/D 转换被存入卡上缓存,写满 1kb 后,即 5.12 ms 后,采集卡发出中断。在 Windows 2000/

XP 的 DPC 机制下,中断响应时间可以达到 $10\mu\text{s}$ 以下。通过高精度计数器的测量,1kb 数据的转移时间最多需要 1ms。数据转移到一级缓存后,核心信号发出,用户程序在下一个时钟中断时能收到信号,即 10ms 以后,开始数据从一级缓存至二级缓存的转移。通过上面的数据,可以看出,转子所产生的振动信号,最多经过 18ms 的延时,就能送到应用程序中进行处理。这样的响应速度,对于转子的故障监测是完全能够满足要求的。

图 2 所示为系统在对某周期性信号进行采集时所绘出的波形图,可以看出该信号的周期性保持良好,没有出现由于漏采而导致的周期不固定现象。同时,波形完整而没有突变,说明采集、存盘和显示线程在缓冲机制下,速度匹配得很好。

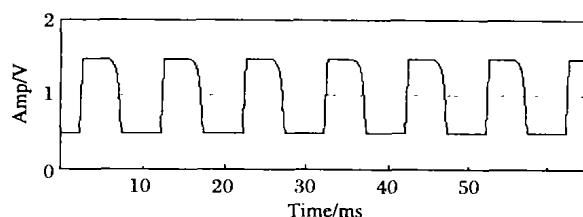


图2 波形图

表2 测定的振动不平衡量

测试次数	1	2	3	4
频率(Hz)	66.02	65.83	66.45	67.08
幅值(μm)	75.08	74.28	75.71	75.37
相位($^\circ$)	31.05	31.35	32.05	31.96

表 2 给出了系统在实验中测出的转子振动不平衡量,可以看出,由于系统的实时性良好,在转子转速有小波动的情况下,仍能根据键相信号的上升沿利用软件实现锁相倍频,实验数据具有很好的重复性。实际动平衡的结果(文中未给出)也证明了本方案具有良好的实用性。

4 结语

在通用操作系统上完成实时应用,最大的障碍就在于它的不确定性。本文所提出的解决方案从这一点着眼,采用多线程、DPC、APC、用户定时器、用户自定义消息、原子操作、核心信号量、中断等多项技术,综合利用了 Windows 2000/XP 为实时应用提供的所有机制,实现了采集卡保持在最高采样频率下的实时数据的完整采集、大容量连续存盘和同步显示与分析。在针对转子实验台动平衡等实验的应用中,无论从实时性方面,还是从所实现的精度方面,都取得很好的效果,为 Windows 2000/XP 平台下,用软件完成以往需要硬件才能完成的实时操作与运算、控制提供了一种较好的解决方案。

参考文献:

- [1] 张李超,等. Windows NT 的实时性研究[J]. 计算机工程与应用, 2002, (5): 41-42.
- [2] 王文武,等. 多媒体定时器的定制和使用方法[J]. 计算机应用, 2000, (3): 39-40.
- [3] RICHTER J. Windows 核心编程[M]. 北京: 机械工业出版社, 2000.
- [4] 罗日成,等. Windows 98 下基于 VxD 技术的中断编程与实现[J]. 计算机工程, 2002, (2): 281-283.