

文章编号 :1007 - 649X(2001)01 - 0052 - 04

Windows 环境下硬件中断的性能分析

徐 峰,王宪平

(国防科技大学机电工程与自动化学院,湖南 长沙 410073)

摘 要:分析了 32 位 Windows 系统实现硬件中断服务的内部机理和系统对中断信号的响应性能,通过比较多种测试环境下得到的中断延时,讨论了中断频率、中断方式、调度方式等因素对 Windows 硬件中断实时响应性能的影响.指出在 32 位 Windows 系统中中断延时不可避免的情况下,采用非共享式中断,以及合适的调度方法,可以相对减少中断延时,提高硬件中断的实时响应性能.

关键词:硬件中断;中断响应;性能分析

中图分类号:TP 303 文献标识码:A

0 引言

中断是微机系统的重要功能之一,当系统出现紧急情况,需要暂停正常程序运行并插入临时处理程序时,计算机能正确地打断主程序,然后转入相应的中断处理程序.对于计算机接口管理来讲,由于有了中断功能,CPU 可以不必经常查询外围接口,空出的时间可去处理其他事,直到接口提出需要服务的中断请求为止.

在一个中断驱动的系统,如现代的 PC 机,Intel 8259A 这样的可编程控制器就是系统的任务主管,它接收从各种中断源发出的中断请求,选出其中优先级最高的一个,然后判断是否高于 CPU 当前正在执行的任务优先级,如果此中断请求优先级较高,则中断 CPU 的工作,执行相应于此中断的服务程序(Interrupt Service Routine).因此,当中断控制器向 CPU 发出中断指令时,还要提供适当的地址等信息,使得 CPU 能够转到此中断相应的中断服务程序的入口处,此地址又称为中断向量.

PC 机系统的中断分为 3 类,即硬中断、异常中断和软件中断.硬中断也称外部中断,它可分为两种,一种是由中断电路发出的中断请求信号在 CPU 的 INTR 端引起的中断,它常常由 PC 机的外设或一些接口功能产生,如键盘、打印机、串行口等等,同时它也是一种可以屏蔽的中断.另外一种

是 CPU 的 NMI 端引起的中断,也称非屏蔽中断,当系统出现掉电、内存奇偶校验错误等,系统使用非屏蔽中断.我们通常处理的都是外部中断.大多数情况下,应用系统对中断响应的性能都有一定要求.如在测量 - 反馈 - 控制的闭环系统中使用硬件中断进行定时,就对定时的准确性和均匀性有着严格的要求.然而,并非所有中断源发出的所有中断请求 CPU 都能马上响应,而是有一定条件的.所谓中断延时是指中断源产生一个中断请求信号与该中断被响应之间的时间间隔.对于 CPU 外部中断源的请求,在正确设置了中断寄存器的情况下仍然受很多因素影响,从而不可避免地导致中断响应存在延迟,甚至会丢失个别信号.在多任务的 Win32 系统中,延迟就更加严重.

1 Windows 实现硬件中断的类型和机理

在实模式操作系统,如 DOS 中,PC 机所有的中断执行都是依靠中断向量表索引来完成调用的.32 位的 Windows 操作系统运行在保护模式下,其硬件中断与实模式相似,都是由外部接口通过中断电路向 CPU 申请中断,其中断向量也是由中断电路提供(NMI 除外).但是处理硬件中断的机理就大不相同了.

在 386 以上 CPU 保护模式下,CPU 仍然是通过中断向量去完成中断服务程序调用的,但其索引过程却是由一个称作中断描述符表 IDT 中的描

收稿日期 2000 - 10 - 10;修订日期 2000 - 11 - 22

作者简介:徐 峰(1972 -)男,湖北省武汉市人,国防科技大学讲师,硕士研究生,主要从事精密工程与计算机控制方面的研究.

万方数据

述符完成的,具体的过程参见文献 [1]。

由于 Windows 9x 和 Windows NT(以及最新的 Windows 2000)具有保护机制,工作在 CPU 的用户态(3 环)的程序不能直接对系统硬件进行操作,所以中断的处理只能在 CPU 的特权态(0 环)进行,而只有操作系统的核心代码可以在 0 环运行,所以中断的处理必须以系统核心的插件形式实现,具体的形式有 VxD、SYS 和 WDM3 种。

VxD 是虚拟设备驱动程序的简称,一个虚拟设备就是一段 32 位可执行代码,作为系统内核的一部分,实现管理系统资源、共享各种硬件设备并在共享的同时协调好硬件中断、I/O 端口读写这样的棘手操作,避免它们可能引起的系统崩溃,系统已经实现了多个虚拟设备,用户的 VxD 可以 DLL 的形式链入 Windows 操作系统的核心层(ring 0),它主要解决不能被 ring 3 层用户态应用程序解决的一系列与硬件有关的问题。VxD 适用于 Windows 3x、Windows 9x,但不能使用在 NT 和 Windows 2000 下。

当 Windows 95 启动时,它初始化中断描述表,中断是由虚拟机管理器(Virtue Machine Manager)处理的。当发生中断时,VMM 和一个叫做虚拟可编程中断控制器(Virtual Programmable Interrupt Controller)的组件协调工作。后者虚拟化 PC 机中的双 8259APIC。发生中断时,VMM 接管控制(即此中断由 VMM 捕获),然后,VMM 通知 VPICD 这个中断事件。希望虚拟此中断的 VxD,必须事先向 VPICD 登记。当 VPICD 接收到某一中断事件时,就会通知登记到此中断的 VxD。

SYS 指 NT 式驱动程序,主要使用在 Windows NT 中,其中一小部分也可以使用在 Windows 2000 中。WDM 是对 NT 式驱动程序的增强,它可以使用在 Windows 98、NT 和 Windows 2000 中,但不能使用在 Windows 95 中。它部分地引入了类的概念,提供了一系列的类驱动程序(Class Drivers)。类驱动程序完成针对一类设备的一系列基本操作。对于这些类的设备,开发者只在自己编写的驱动程序中调用合适的类驱动程序,而不需要写一个核模式的驱动程序。

2 Windows 中影响硬件中断性能的因素

Windows 中影响硬件中断性能主要有 4 个方面的因素。

(1) PC 系统中,操作系统并不是在外部硬件电路产生中断请求信号后就立即进入中断处理程

序,下面分析系统操作软件和中断服务程序响应硬件中断的步骤。

CPU 在现行指令结束后,即运行到最后一个机器周期的最后一个 T 状态时,才能采样 INTR 线而响应可能提出的外中断请求。如发现有中断请求,则将 CPU 内部的中断锁存器置 1,然后下一个机器周期进入中断周期,而不进入新的取指周期。此时,CPU 就响应中断请求,并自动关中断,然后进入为之服务的中断处理程序。在中断处理程序中,应该先后完成以下几项工作。①保护现场。为使中断处理程序不影响被中断程序运行,必须首先将断点处的有关各寄存器的内容和标志位的状态压入堆栈保护起来,以备中断处理完毕后能返回原程序,从断点开始正确执行。②开中断。以便执行中断服务程序时,能响应更高级别的中断源请求。③完成输入输出或异常事件处理的服务程序。这是中断处理程序的核心。④关中断。保证在恢复现场时不被新的中断所打扰。⑤恢复现场。中断服务程序结束后,必须进行现场恢复的操作。多数 CPU 是用 POP 指令把保存的断点信息从堆栈中弹出,达到恢复现场的目的。⑥开中断。以便中断返回后可响应新的中断请求。⑦中断返回。最后必须安排一条中断返回指令,使断点地址送回程序计数器或 CS:IP,继续执行被中断的程序。

可见,为了完成中断服务程序的真正工作(第 3 步)必须实现上述这些步骤,这就不可避免消耗 CPU 的时间。

(2) 在多中断源的中断系统中,同一时刻只能响应一个中断请求,当有较高优先级的中断请求存在或被响应时,较低优先级的中断请求不能被响应。这时,中断的延迟就更大了,而且难以估计。中断延时决定了硬件产生中断并可以得到正确响应的最高频率。如果硬件产生中断的频率高于此频率,则系统肯定工作不正常,甚至死机。

如果在 Windows 98 中的“开始 | 设置 | 控制面板 | 系统 | 设备管理器”一栏中双击最上面的“计算机”,然后查看中断请求 IRQ,会发现一个 IRQ 上可能有多个设备,这就是中断共享。它是为了解决中断资源不够的情况而提出来的。但中断共享会造成系统硬件中断响应的不确定性,甚至于死机。

考虑通常采用的边缘触发中断方式的情况,系统运行挂在此中断向量上的所有中断服务程序,希望其中的一个会正确处理。如果设备在发出中断请求时,没有给出相关信息来说明中断发生

的原因或要求中断的具体设备 ,那么情况就很糟了.结果 ,操作系统可能会在检测到中断信号与处理这个中断之间漏掉外部的中断请求信号.

(3) 由于 0 环的中断服务程序(VxD ,SYS , WDM)可以完成的工作受很大的限制 ,如无法进行浮点运算等 ,因此中断服务程序通常采用同步对象的办法 ,通知 3 环的应用程序.但是 Intel 系列 CPU 花费在 0 环和 3 环切换上的代价是非常大的 ,所以中断延时提高了几个数量级以上.

(4) Windows 9x 会在系统的一些关键时刻禁止中断 ,否则可能会导致不确定的状态 ,如打开一个 DOS 窗口时 ,VMM 必须创建一个局部描述符表(LDT) ,一个任务状态段(TSS) 和一个分页表(Page Table) 并执行一系列的整理工作.如果此时发生中断 ,系统将会死机.这也是 Windows 9x 这样的 16 位和 32 位代码混合系统所带来的问题.因此 , Windows 9x 在这些时刻 ,就不得不使用 CLI 和 STI 指令.令人遗憾的是 ,在 Windows 9x 中这样的关键时刻并不少^[1].

3 实验数据和分析

在 Windows 操作系统中 ,准确测量中断的延时是非常困难的.使用高精度的定时可以获得未必准确但却是有用的结果.一个 60 MHz 的 Pentium 处理器的一个指令周期大概为 16 ns ,因此定时器不能测量几十个指令的时间 ,但可以大致测量几千、几万个指令的时间.现在的 CPU 速度已经可以到 1GHz 了 ,中断延时的测量就更加困难^[1].

当外部电路产生中断时 ,先通过硬件打开定时器 ,在中断服务程序中关闭定时器 ,这样就可以大致地测量到中断的延时 ,可以通过多次测量再平均 ,以提高可信度.通过测试环境的改变 ,来测定出影响中断延时的因素.下面是在一台 Pentium 60MHz ,32MB DRAM ,运行 Windows 95 的计算机上 ,对下列 4 种情况下的中断延时进行测试所得的数据 ,我们就在此基础上进行分析.①系统空闲.②运行 WINBEZMT. EXE 的 4 个实例 ,每个实例有 4 个线程. WINBEZMT. EXE 是一个多文档界面(MDI)应用程序 ,每个子窗口有自己的线程 ,使用 Win32 API PolyBezier() 不停地画 Bezier 曲线 ,这是一个大量占用 CPU 和 GDI 的程序.③运行 MS - DOS 应用程序 DOSFP. EXE 的 4 个实例 ,此程序执行一个无限循环 ,不停地进

行浮点除法运算.④运行 MS - DOS 应用程序 DOSIO. EXE 的 4 个实

例 ,此程序执行一个无限循环 ,不停地打开、读和关闭一个 100KB 的文件 ,缓冲区为 1KB. Windows 9x 会很快地缓存这个文件 ,但是 DOSIO. EXE 产生了许多软件中断.

表 1 测量到的中断延时^[1]

Table 1 The measured interrupt response delay				
测量环境	空闲	WINBEZMT	DOSFP	DOSIO
平均中断延时/ μ s	11	21	11	15

空闲状态下 ,中断延时为 11 μ s ;运行 DOSFP 时的中断延时也是 11 μ s ,表明运行计算密集些任务的虚拟机对于中断延时没有什么影响.运行 DOSIO 时导致了中断延时的一些增加 ,原因是由于 DOSIO 不停产生软件中断.而运行 WINBEZMT 时的中断延时是比较大的.用 SoftICE 对程序跟踪 ,发现它的每一个子线程都是一个不断的循环 ,在调用 PolyBezier()后 ,又调用 Sleep()进行 1ms 的延时.使用 SoftICE 跟踪 Sleep()的代码发现 ,KERNEL32. DLL 中的 Sleep()函数实际上通过 INT 30h 保护模式回调机制调用 VWIN32. VXD 实现的. VMM 提供了 Allocate _ PM _ Callback 服务 ,实现这种回调机制 ,这样 KERNEL32. DLL 这样的保护模式组件就可以调用 VWIN32. VXD 这样的 0 环代码.当 INT 30h 指令执行时 ,中断被禁止 ,控制转移到 VMM. VMM 检查 INT 30h 的地址 ,以确定合适的 0 环代码句柄.在 Sleep()中 ,VWIN32. VXD 中的代码实际实现了 Sleep()函数的功能.然后 ,重新允许中断.在 WINBEZMT. EXE 运行过程中 ,产生了大量的 INT 30h 中断 ,从而导致外部中断的响应延时.即使在 Windows NT 这样的纯 32 位操作系统中 ,某些运行机制也会导致中断延迟的巨大增加.

前面提到 ,设备驱动程序中可以完成的任务受到很大的限制 ,必须采取某种方法通知应用程序. Windows 95 提供了 3 种主要的机制完成上述的调度工作 :全局事件、虚拟机事件和优先虚拟机事件.下面的数据表明了不同的调度机制对中断延时的影响.

全局事件调度方法允许 Windows 95 在中断环境外的任何虚拟机环境中执行这些延迟实现代码(DPC).这样 ,中断延时相对较小 ,如表 2 所示.

但由于 VxD 驱动程序通常都有相应的 3 环驱动程序代码执行系统虚拟机的工作 ,因此 ,在这种情况下 ,全局事件调度机制的作用不大.

另外一种调度 DPC 代码的方法是使用虚拟机事件 ,这种调度告诉 Windows 95 在特定的虚拟

机环境下执行 DPC 代码.如果当前的虚拟机不是指定的目标虚拟机,就会产生额外的中断延时,因为 DPC 代码将会推迟到当前虚拟机是指定虚拟机的情况下才会执行.表 3 的 DOSFP 和 DOSIO 显示了这种情况,此时指定的虚拟机是系统虚拟机,而当前执行的是 DOSFP 和 DOSIO 虚拟机,所以会发生非常大的延时.

表 2 使用全局事件调度时的中断延时^[1]

Table 2 The interrupt response delay in global event dispatching				
测量环境	空闲	WINBEZMT	DOSFP	DOSIO
平均中断延时/ μ s	43	64	29	66

表 3 使用 VM 事件调度时的中断延时^[1]

Table 3 The interrupt response delay in VM event dispatching				
测量环境	空闲	WINBEZMT	DOSFP	DOSIO
平均中断延时/ μ s	45	65	9291	9191

最后一种调度延迟过程调用 DPC 代码的方法是使用优先虚拟机事件,VMM 会提高指定的虚拟机的优先级,从而减小中断延时,如表 4 所示.显然,如果 VxD 驱动程序既需要驱动较高带宽的硬件,又需要和系统虚拟机的 3 环代码合作,采用这种调度方法可以获得相对较小的中断延时.

表 4 使用优先 VM 事件调度时的中断延时^[1]

Table 4 The interrupt response delay in priority VM event dispatching				
测量环境	空闲	WINBEZMT	DOSFP	DOSIO
平均中断延时/ μ s	60	100	192	280

4 结论

(1)32 位 Windows 操作系统下,实现硬件中断必须采用 0 环的设备驱动程序.而且,由于 PC 硬件、操作系统软件和应用软件各方面的原因,中断的延迟是不可避免的.由于这个原因,32 位 Windows 操作系统下硬件中断的频率不可能做到很高,大约在十几 kHz 左右.

(2)0 环的设备驱动程序可以完成的任务受到限制,因此必须采取同步对象、DPC 等办法异步地完成操作.

(3)在一定的情况下,可以一定程度的减小中断延迟,保证一定的中断均匀性.显然,高频率的 CPU 对中断性能改善较大.

(4)因为 PC 系统中硬件中断资源有限,32 位 Windows 操作系统为解决这个问题而采取中断共享方式会导致较大的中断延迟.因此,基于实时性的考虑,用户在开发自己的设备驱动程序时,要尽量避免这种情况.

参考文献：

[1] 李圣怡. Windows 环境下的软硬件接口技术[M]. 长沙 :国防科技大学出版社,2000.
[2] 李广军,王厚军.实用接口技术[M].成都 :电子科技大学出版社,1998.
[3] CHRIS Cant. Windows WDM 设备驱动程序开发指南[M].北京 :机械工业出版社,2000.
[4] 杨 强,李堂秋. Win 9x 虚拟设备驱动程序编程指南[M].北京 :清华大学出版社,1999.

The Analysis of the Performance of Hardware Interrupt in Windows Environment

XU Feng , WANG Xian - ping

(College of Mechanical Engineering and Automation ,The National University of Defence Technology ,Changsha 410073 ,China)

Abstract :This paper analyses the hardware interruptions disposing mechanism and the interruption response performance of 32 - bit Windows platform ,and the data reflecting based on the comparison among Windows interrupt response delays under diferent environments , the frequence of interruption and the mode of interruption and other factors which have influence on the response performance of the hardware interruptions in 32 bit Windows environment are discussed. Finally , the paper points out that although the interruption response delay in 32 bit Windows enviromment can 't be avoided , adopting non - shared interruption and other appropriate methods can relatively reduce the interrupt response delay and improve the pomfomence of the hardware interruption real time response.

Key words hardware interruption ; interruption response ; analysis of performance