Windows 实时软件的开发方法

Design Methods for Windows Real – time Software

刘 玉 （廊坊师院 数学与信息科学学院 河北 固安 065505）

摘要：本文分析了软件的实时性和不同操作系统对实时性的支持能力，介绍了在 Microsoft Windows 平台上开发具有高度实时性能应用软件的方法。

关键词：实时软件 实时响应 定时器

1 实时性软件的基本要求

实时软件的实时性能直接关系到软件自身的技术内涵和市场价值。实时软件系统的特点在于：一个正确的运行不仅取决于结果的准确，更取决于实现的时间。实时性的衡量标准是系统的时间响应特性，不是系统的平均响应时间而是最坏情况下的响应时间。实时系统有时被进一步划分为"硬实时"系统和"软实时"系统。一个典型的"硬实时"系统的案例是压盖机给在传送带上传送的瓶子加盖，对于该系统，仅仅准确定位压盖机是不够的，如果瓶子已经移走而压盖机才到位，那么所有的精确定位都将是徒劳的。除了可靠性，实时系统通常还有一些其他要求，主要是：

（1）快速的时钟和定时器；

（2）一个具有足够优先级的多线程优先级调度器；

（3）可预测的线程同步机制。

2 实时性的涵义

（1）均匀触发。是指每隔一定的时钟周期，系统都会准确触发和调用与时钟周期相关的处理函数。在一定的时间精度下，系统要保证每隔一定的时钟周期准时发生触发。

（2）实时响应。是指在系统向我们的时钟处理函数准时产生触发之后，无论系统的软硬件环境的繁忙程度如何，应用程序的时钟处理函数能够实时地作出响应和运行。

3 操作系统对应用软件实时性的支持

三类比较典型的操作系统对实时性的支持：

（1）独占式操作系统。系统在同一时刻只执行一个任务，所有的软硬件资源都被当前任务独占。这种操作系统的实时性是最好的，比较典型的独占式操作系统如 DOS。

（2）抢占式操作系统。操作系统的内核是抢占式的，支持丰富的进程/线程的优先级别，同时拥有精确的调度机制。这种操作系统的实时性支持是非常优秀的，比较典型的抢占式操作系统有：Linux 2.6 版内核、Windows CE 4.2 版本以上等。

（3）非抢占式操作系统。操作系统对任务的执行往往采用轮询的方式，系统支持的优先级匮乏，而且对线程的调度也不可预测，比较典型的操作系统有：Windows 9X / Me / NT / 2K / XP / XP Embedded 等。

在非抢占式操作系统上开发实时软件，可通过对操作系统扩展的方式，使其具有实时特性。如美国 Ardence 公司的 RTX 产品，该产品位于 Windows 的 HAL（硬件抽象层），在 Ring 0 级别向 Ring 3 层的应用程序提供精确的时钟触发和线程优先级调度管理，实现了精确度为 100 纳秒的时钟周期和 64 种线程优先级别及其调度机制。

4 Windows NT / 2000 / XP / XP Embedded 环境下的实时性开发

4.1 定时器的选择

Windows 提供以下三种定时器模型以供选择：系统定时器、事件定时器和多媒体定时器。

4.1.1 系统定时器

Windows 系统向应用开发提供一种基于消息的系统定时器，在初始化系统定时器时，设置时钟触发的周期的数值（单位为毫秒，以下同），然后 Windows 将按照这个数值周期性地向当前应用程序发送“WM_TIMER”的消息，应用程序只需响应这个系统消息。

经测试，系统定时器的最小触发周期为 55 毫秒，Windows 无法实现更低的时钟周期。而且系统定时器的抗干扰性很弱，即在其它进程或线程大量占用 CPU 的情况下，系统定时器将很难保持稳定的触发周期。

4.1.2 事件定时器

Windows 为异步编程提供了事件机制。利用事件句柄和 Windows 提供的系列等待 API 函数，可以编写一个基于事件的定时器模型，其示例如下：

```
while ( ::WaitForSingleObject( hEvent, 1 ) == WAIT_TIMEOUT )
    {
      // 加入需要周期执行的代码 ;
    }
```

其中，“WaitForSingleObject( )”是 Windows 提供的用于等待处理的 API 函数，它的第一个参数是对象的句柄，此对象可以是事件或者线程等等，在上例中，是一个事件的句柄；第二个参数是函数等待的时间，以毫秒为单位，如在指定的时间内对象未被调用，则该函数立即返回；这个 API 有三个返回值：

（1）WAIT_ABANDONED：表示该函数等待的对象被释放了。

（2）WAIT_OBJECT_0：表示该函数等待的对象被激活、设置或调用了。

（3）WAIT_TIMEOUT：表示该函数等待的对象在指定时间内未被调用，超时了。

经测试，事件定时器的最小触发周期为 20 毫秒，Windows 无法实现低于这个数值的时钟周期。和系统定时器一样，事件定时器的抗干扰性也很弱。

4.1.3 多媒体定时器

自从 Microsoft 发布 Windows 95 以后，就给 Windows 增添了用于支持多媒体的动态链接库—“WinMM.dll”，除了通用的多媒体功能支持外，还提供了一个多媒体定时器。对它的操作由一系列特殊的 API 构成，其中最核心的是 timeSetEvent( ) 函数，其函数原型为：

```
MMRESULT timeSetEvent( UINT uDelay,
UINT uResolution,
LPTIMECALLBACK lpTimeProc,
DWORD_PTR dwUser,
UINT fuEvent );
```

经测试，多媒体定时器的最小触发周期能够精确到 1 毫秒，精度非常之高，而且抗干扰能力相对较强，是 Windows 实时软件开发中理想的定时器。

4.2 实时响应能力

对于实时软件开发来说，找到了优秀的定时器还不够，操作系统所提供的实时响应能力才是至关重要的。Windows NT / 2000 / XP / XP Embedded 平台都具有以下特点：

（1）线程优先级太少，只有七种；

（2）隐含的不确定的线程调度机制；

（3）优先级倒置，尤其体现在中断处理中。

4.3 系统的实时性能测试方案与结果分析

（1）测试方案。测试程序分别采用同步/异步的多媒体定时器产生周期触发，其中异步定时器分别提供两种优先级（最高优先级别和缺省优先级别）。测试程序每 10 毫秒产生一次时钟周期触发，每次触发记录下相邻两次触发的时间间隔（单位为毫秒），每产生一千个记录数据，将其保存到文件中，以备以后对测试结果的分析。在测试程序运行的同时，在后台用资源管理器复制大数据量的文件，人为的造成“File Flush”的效果对其执行进行干扰。

（2）测试结论。除非借助于第三方的硬实时插件，在 Windows NT / 2000 / XP / XP Embedded 平台上，很难实现具有硬实时特性的应用。

5 Windows CE 环境下的实时性开发

Windows CE（WinCE 约定为 WinCE 4.2 版本以上）是 Microsoft 推出的专门面向嵌入式应用的操作系统。WinCE 是一个优秀的实时性应用平台，它具备以下特点：

（1）丰富的线程优先级，多达 256 级；

（2）精确的线程调度机制。

5.1 一个特殊的 API

从 WinCE 3.0 起，CE 开始支持 256 个级别的线程

优先级。具体的 API 函数如下：

```
BOOL CeSetThreadPriority( HANDLE hThread,
    int nPriority );
```

(1) hThread：线程句柄；

(2) nPriority：线程优先级。有效值从 0 至 255，其中 0 的优先级最大，并向后优先级逐一减小。优先级共分以下几类：

- 0 到 96：为实时应用保留的高于设备驱动的优先级；
- 97 到 152：缺省的 WinCE 系统的设备驱动的优先级；
- 153 到 247：为实时应用保留的低于设备驱动的优先级；
- 248 到 255：非实时应用的优先级。

(3) 返回值：声名成功返回 TRUE，失败返回 FALSE。可以调用 GetLastError( ) 查询错误码。

5.2 测试方案与结果分析

(1) 测试方案。测试程序运行在 WinCE 4.2 下，每 10 毫秒产生一次时钟周期触发，每次触发记录下相邻两次触发的时间间隔(单位为毫秒)，每产生一千个记录数据，将其保存到文件中，以备以后对测试结果的分析。

在测试程序运行的同时，在后台用 CE 的资源管理器复制大数据量的文件，人为的造成"File Flush"的效果对其执行进行干扰。

(2) 测试结果及分析

① 没有实时性控制的测试结果：对于没有实时性控制的应用，当后台发生"File Flush"时，其测试结果如图 1：不难发现，当系统忙碌时，当前应用线程的实时性无法保证。

② 加入实时性控制的测试结果：对于加入实时性控制的应用，当后台发生"File Flush"时，其测试结果如图 2：可以发现，当系统忙碌时，当前应用线程具有优秀的实时性，相邻两次周期触发的执行一直平稳地保持在实现设定的 10 毫秒。

③ 测试结论：在 WinCE 4.2 系统下，通过采用异步的多媒体定时器和线程的高优先级设置，能够实现达到"硬实时"标准的实时性。

6 结束语

综上可见，在 Windows NT / 2000 / XP / XP Embedded 平台上，除非借助于第三方的硬实时插件之外，很难实现具有硬实时特性的应用；而在 WinCE 平台，通过采用异步的多媒体定时器和线程的高优先级

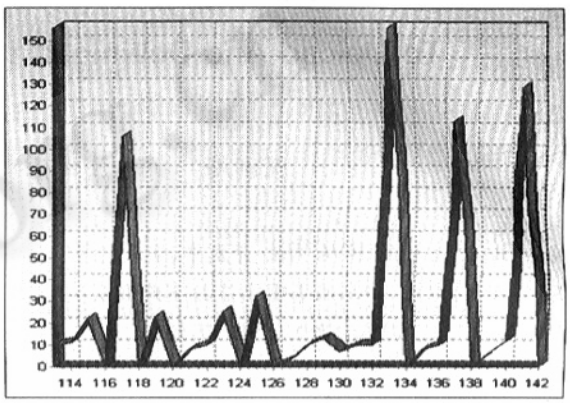

图 1　没有实时性控制的测试

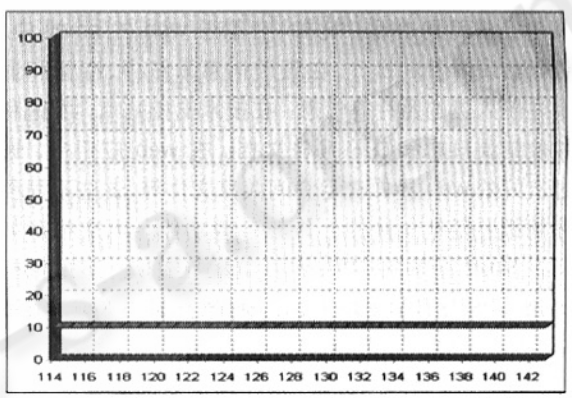

图 2　加入实时性控制

设置，能够实现达到"硬实时"标准的实时性。在 Windows 应用开发中，WinCE 是理想的实时应用开发平台。高优先级带来了实时性能的大幅度提高，也对相关开发带来了更高的要求：一定要确保高优先级的线程的执行是稳定的、高效的和非阻塞的，否则，这将给 WinCE 系统的性能带来巨大的伤害。