**[QueryPerformanceFrequency用法](http://www.cppblog.com/deane/articles/113151.html)**

**精确获取时间**

QueryPerformanceFrequency() - 基本介绍

类型：Win32API

原型：BOOL QueryPerformanceFrequency(LARGE\_INTEGER \*lpFrequency);

作用：返回硬件支持的高精度计数器的频率。

返回值：非0，硬件支持高精度计数器；0，硬件不支持，读取失败。

QueryPerformanceFrequency() - 技术特点

供WIN9X使用的高精度定时器：QueryPerformanceFrequency()和QueryPerformanceCounter()，要求计算机从硬件上支持高精度定时器。需包含windows.h头文件。

函数的原形是：

BOOL QueryPerformanceFrequency(LARGE\_INTEGER \*lpFrequency);

BOOL QueryPerformanceCounter (LARGE\_INTEGER \*lpCount);

数据类型LARGEINTEGER既可以是一个作为8字节长的整数，也可以是作为两个4字节长的整数的联合结构，其具体用法根据编译器是否支持64位而定。该类型的定义如下：

typeef union \_ LARGE\_INTEGER

{

struct

{

DWORD LowPart;

LONG HighPart;

};

LONGLONG QuadPart;

} LARGE\_INTEGER;

在定时前应该先调用QueryPerformanceFrequency()函数获得机器内部计时器的时钟频率。接着在需要严格计时的事件发生前和发生之后分别调用QueryPerformanceCounter()，利用两次获得的计数之差和时钟频率，就可以计算出事件经历的精确时间。

测试Sleep的精确时间：

#include <stdio.h>

#include <windows.h>

void main()

{

     LARGE\_INTEGER nFreq;

     LARGE\_INTEGER nBeginTime;

     LARGE\_INTEGER nEndTime;

     double time;

     QueryPerformanceFrequency(&nFreq);

     QueryPerformanceCounter(&nBeginTime);

     Sleep(1000);

     QueryPerformanceCounter(&nEndTime);

     time=(double)(nEndTime.QuadPart-nBeginTime.QuadPart)/(double)nFreq.QuadPart;

     printf("%f\n",time);

}

结果为

0.999982

1.000088

1.000200

等，所以Sleep的精度还是比较低的。