C++ clock()函数主要是帮助我们实现计时的功能。而与其相关的数据类型是clock\_t。在MSDN中，查得对clock函数定义如下：

clock\_t clock( void );

这个函数返回从“开启这个程序进程”到“程序中调用C++ clock()函数”时之间的CPU时钟计时单元（clock tick）数，在MSDN中称之为挂钟时间（wal-clock）。其中clock\_t是用来保存时间的数据类型，在time.h文件中，我们可以找到对 它的定义：

#ifndef \_CLOCK\_T\_DEFINED

typedef long clock\_t;

#define \_CLOCK\_T\_DEFINED

#endif

很明显，clock\_t是一个长整形数。在time.h文件中，还定义了一个常量CLOCKS\_PER\_SEC，它用来表示一秒钟会有多少个时钟计时单元，其定义如下：

#define CLOCKS\_PER\_SEC ((clock\_t)1000)

可以看到每过千分之一秒（1毫秒），调用C++ clock()函数返回的值就加1。下面举个例子，你可以使用公式clock()/CLOCKS\_PER\_SEC来计算一个进程自身的运行时间：

void elapsed\_time()

{

printf("Elapsed time:%u secs."n",clock()/CLOCKS\_PER\_SEC);

}

当然，你也可以用clock函数来计算你的机器运行一个循环或者处理其它事件到底花了多少时间：

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int main( void )

{

long i = 10000000L;

clock\_t start, finish;

double duration;

/\* 测量一个事件持续的时间\*/

printf( "Time to do %ld empty loops is ", i );

start = clock();

while( i-- ) ;

finish = clock();

duration = (double)(finish - start) / CLOCKS\_PER\_SEC;

printf( "%f seconds\n", duration );

system("pause");

return 0;

}运行结果如下：

Time to do 10000000 empty loops is 0.03000 seconds

以上就是对C++ clock()函数的相关应用介绍。上面我们看到时钟计时单元的长度为1毫秒，那么计时的精度也为1毫秒，那么我们可不可以通过改变CLOCKS\_PER\_SEC的定义，通过把它定义的大一 些，从而使计时精度更高呢？通过尝试，你会发现这样是不行的。在标准C/C++中，最小的计时单位是一毫秒。