**C语言time.h中clock()函数的使用**

C语言中求程序执行的时间可以使用clock()函数，这个函数返回从“开启这个程序进程”到“程序中调用clock()函数”时之间的CPU时钟计时单元（clock tick）数，其类型为clock\_t，在time.h文件中，可以找到对clock\_t类型的定义：

#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clock（）函数返回的值就加1。  
下面给出一个示例：

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>
4. **int** main()
5. {
6. **long** i=10000000L;
7. **clock\_t** start,finish;
8. **double** TheTimes;
9. printf("做%ld次空循环需要的时间为",i);
10. start=clock();
11. **while**(i--);
12. finish=clock();
13. TheTimes=(**double**)((finish-start)/CLOCKS\_PER\_SEC);
14. printf("%f秒。\n",TheTimes);
15. **return** 0;
16. }

可是运行完每次的结果都为：

1. 做10000000次空循环需要的时间为0.000000秒。
2. Process returned 0 (0x0) execution time : 0.438 s
3. Press any key to continue.

加大循环次数也不行。

最后终于找到的问题所在：

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>
4. **int** main()
5. {
6. **long** i=10000000L;
7. **clock\_t** start,finish;
8. **double** TheTimes;
9. printf("做%ld次空循环需要的时间为",i);
10. start=clock();
11. **while**(i--);
12. finish=clock();
13. TheTimes=(**double**)(finish-start)/CLOCKS\_PER\_SEC;
14. printf("%f秒。\n",TheTimes);
15. **return** 0;
16. }

看红色行和上面的区别，现在明白了吧！是由于括号运算符改变了运算的优先级，将里面的四则运算变成了整数之间的除法，造成位数被舍去，造成的误差。