<http://blog.chinaunix.net/uid-20548989-id-2533161.html>

一、struct timeval结构体

struct timeval结构体在time.h中的定义为：

1. struct timeval
2. {
3. \_\_time\_t tv\_sec;        /\* Seconds. \*/
4. \_\_suseconds\_t tv\_usec;  /\* Microseconds. \*/
5. };

其中，tv\_sec为Epoch到创建struct timeval时的秒数，tv\_usec为微秒数，即秒后面的零头。比如当前我写博文时的tv\_sec为1244770435，tv\_usec为442388，即当前时间距Epoch时间1244770435秒，442388微秒。需要注意的是，因为循环过程，新建结构体变量等过程需消耗部分时间，我们作下面的运算时会得到如下结果：

1. #include <sys/time.h>
2. #include <stdio.h>
4. int
5. main(void)
6. {
7. int i;
8. struct timeval tv;
9. for(i = 0; i < 4; i++){
10. gettimeofday(&tv, NULL);
11. printf("%d\t%d\n", tv.tv\_usec, tv.tv\_sec);
12. sleep(1);
13. }
14. return 0;
15. }
16. 329612 1314851429
17. 329782 1314851430
18. 329911 1314851431
19. 330036 1314851432

前面为微秒数，后面为秒数，可以看出，在这个简单运算中，只能精确到小数点后面一到两位，或者可以看出，每进行一次循环，均需花费0.005秒的时间，用这个程序来作计时器显然是不行的，除非精确计算产生的代码消耗时间。

二、gettimeofday()函数

原型：

1. /\* Get the current time of day and timezone information,
2. putting it into \*TV and \*TZ. If TZ is NULL, \*TZ is not filled.
3. Returns 0 on success, -1 on errors.
4. NOTE: This form of timezone information is obsolete.
5. Use the functions and variables declared in <time.h> instead. \*/
6. extern int gettimeofday (struct timeval \*\_\_restrict \_\_tv,
7. \_\_timezone\_ptr\_t \_\_tz) \_\_THROW \_\_nonnull ((1));

gettimeofday()功能是得到当前时间和时区，分别写到tv和tz中，如果tz为NULL则不向tz写入。