一、linux平台下的计时函数gettimeofday

int gettimeofday ( struct timeval \* tv , struct timezone \* tz ),gettimeofday()会把目前的时间有tv所指的结构返回，当地时区的信息则放到tz所指的结构中.

//timeval结构定义为:

struct timeval{

long tv\_sec; /\*秒\*/

long tv\_usec; /\*微秒\*/

};

//timezone 结构定义为:

struct timezone{

int tz\_minuteswest; /\*和Greenwich 时间差了多少分钟\*/

int tz\_dsttime; /\*日光节约时间的状态\*/

};

void test7()

{

struct timeval t1,t2;

double timeuse;

gettimeofday(&t1,NULL);

foo();

gettimeofday(&t2,NULL);

timeuse = t2.tv\_sec - t1.tv\_sec + (t2.tv\_usec - t1.tv\_usec)/1000000.0;

printf("Use Time:%f s\n",timeuse);

}

Test file：

注：

在VC中，Sleep()里面的单位，是以毫秒为单位，所以如果想让函数滞留1秒的话，应该是Sleep(1000); 在Linux下,sleep()里面的单位是秒，而不是毫秒。

#include <stdio.h>

#include <sys/time.h>

int main()

{

double time\_use=0;

struct timeval start;

struct timeval end;

gettimeofday(&start,NULL);

int i = 100000000;

while(i--);

gettimeofday(&end,NULL);

time\_use=(end.tv\_sec-start.tv\_sec)\*1000000+(end.tv\_usec-start.tv\_usec);//微秒

time\_use/=1000.0;

printf("time\_use is %.f ms\n",time\_use);

}

在windows下的Sleep()函数需要包含windows.h头文件，而在Linux下需要包含的头文件是unistd.h头文件，说明sleep()函数不是标准的C语言库，而且在Windows下Sleep()睡眠时间为毫秒，而Linux下的sleep()函数时间为秒，

---------------------

作者：ShomyLiu

来源：CSDN

原文：https://blog.csdn.net/shomy\_liu/article/details/45110949

版权声明：本文为博主原创文章，转载请附上博文链接！

double start, end;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Barrier(MPI\_COMM\_WORLD); /\* IMPORTANT \*/

start = MPI\_Wtime();

/\* ... do work ... \*/

MPI\_Barrier(MPI\_COMM\_WORLD); /\* IMPORTANT \*/

end = MPI\_Wtime();

MPI\_Finalize();

if (rank == 0) { /\* use time on master node \*/

printf("Runtime = %f s\n", end-start);

}