# linux中的时间编程

## 时间类型

UTC：世界标准时间，也就是格林威治标准时间（GMT）。

Calendar Time：日历时间，是用“从一个标准时间点（如：1970年1月1日0点）到此时经过的秒数”来表示时间。这个标准时间点对不同的[编译器](http://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)来说会有所不同，但对一个[编译系统](http://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E7%B3%BB%E7%BB%9F)来说，这个标准时间点是不变的，该编译系统中的时间对应的日历时间都通过该标准时间点来衡量，所以可以说日历时间是“相对时间”，但是无论你在哪一个时区，在同一时刻对同一个标准时间点来说，日历时间都是一样的。在C语言中通过time函数获得日历时间。

## 时间获取

#include<time.h>

time\_t time(time\_t \*tloc)

功能：获取日历时间

int gettimeofday(struct timeval \*tv,struct timezone \*tz)

功能：获取从今日凌晨到现在的时间差，常用于计算事件耗时。

struct timeval{

int tv\_sec;//秒数

int tv\_usec;//微秒数

## 时间转换

struct \*gntime(const time\_t \*temp)

功能：将日历时间转化为格林威治标准时间，并保存至TM结构。

struct tm \*localtime(const time\_t \*timep)

功能：将日历时间转化为本地时间

struct tm{

int tm\_sec;

int tm\_min;

int tm\_hour;

int tm\_mday;

int tm\_monl

int tm\_year;

int tm\_wday //本周第几日;

int tm\_yday;//本年第几日

int tm\_isdat;//日光节约时间

};

## 时间显示

char \*asctime(const struct tm \*tm)

功能：将tm格式的时间转化为字符串

char \*ctime(const time\_t \*timep)

功能：将日历时间转化为本地时间的字符串形式

## 延时执行

unsigned int sleep(unsigned int seconds)

功能:是程序睡眠seconds秒。

void usleep(unsigned long usec)

功能：是程序睡眠usec微妙。