<https://blog.csdn.net/weiyuefei/article/details/52959462>

Reference: <http://net.pku.edu.cn/~yhf/linux_c/function/04.html>

asctime（将时间和日期以字符串格式表示）    
相关函数

time，ctime，gmtime，localtime  
   
表头文件

#include  
   
定义函数

char \* asctime(const struct tm \* timeptr);  
   
函数说明

asctime()将参数timeptr所指的tm结构中的信息转

换成真实世界所使用的时间日期表示方法，然后将结果以字

符串形态返回。此函数已经由时区转换成当地时间，字符串

格式为:

“Wed Jun 30 21:49:08 1993\n”  
   
返回值

返回一字符串表示目前当地的时间日期。

附加说明

若再调用相关的时间日期函数，此字符串可能会被破坏。此函数与ctime不同处在于传入的参数是不同的结构。  
  
范例

1. #include <time.h>
2. main()
3. {
4. time\_t timep;
5. time (&timep);
6. printf(“%s”,asctime(gmtime(&timep)));
7. }

执行  Sat Oct 28 02:10:06 2000  
  
ctime（将时间和日期以字符串格式表示）    
相关函数

time，asctime，gmtime，localtime  
   
表头文件

#include  
   
定义函数

char \*ctime(const time\_t \*timep);  
   
函数说明

ctime()将参数timep所指的time\_t结构中的信息转

换成真实世界所使用的时间日期表示方法，然后将结果以字

符串形态返回。此函数已经由时区转换成当地时间，字符串

格式为“Wed Jun 30 21 :49 :08 1993\n”。若再调用相关的

时间日期函数，此字符串可能会被破坏。  
   
返回值

返回一字符串表示目前当地的时间日期。  
   
范例

1. #include<time.h>
2. main()
3. {
4. time\_t timep;
5. time (&timep);
6. printf(“%s”,ctime(&timep));
7. }

执行  Sat Oct 28 10 : 12 : 05 2000  
   
  
gettimeofday（取得目前的时间）    
相关函数

time，ctime，ftime，settimeofday  
   
表头文件

#include   
#include   
   
定义函数

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

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\*/  
};  
上述两个结构都定义在/usr/include/sys/time.h。tz\_dsttime

所代表的状态如下  
DST\_NONE /\*不使用\*/  
DST\_USA /\*美国\*/  
DST\_AUST /\*澳洲\*/  
DST\_WET /\*西欧\*/  
DST\_MET /\*中欧\*/  
DST\_EET /\*东欧\*/  
DST\_CAN /\*加拿大\*/  
DST\_GB /\*大不列颠\*/  
DST\_RUM /\*罗马尼亚\*/  
DST\_TUR /\*土耳其\*/  
DST\_AUSTALT /\*澳洲（1986年以后）\*/  
   
返回值

成功则返回0，失败返回－1，错误代码存于errno。

附加说明EFAULT指针tv和tz所指的内存空间超出存取权限

。  
   
范例

1. #include<sys/time.h>
2. #include<unistd.h>
3. main(){
4. struct timeval tv;
5. struct timezone tz;
6. gettimeofday (&tv , &tz);
7. printf(“tv\_sec; %d\n”, tv,.tv\_sec) ;
8. printf(“tv\_usec; %d\n”,tv.tv\_usec);
9. printf(“tz\_minuteswest; %d\n”, tz.tz\_minuteswest);
10. printf(“tz\_dsttime, %d\n”,tz.tz\_dsttime);
11. }

执行  tv\_sec: 974857339  
tv\_usec:136996  
tz\_minuteswest:-540  
tz\_dsttime:0  
 

gmtime（取得目前时间和日期）    
相关函数

time,asctime,ctime,localtime  
   
表头文件

#include  
   
定义函数

struct tm\*gmtime(const time\_t\*timep);  
   
函数说明

gmtime()将参数timep 所指的time\_t 结构中的信息

转换成真实世界所使用的时间日期表示方法，然后将结果由

结构tm返回。  
结构tm的定义为  
struct tm  
{  
int tm\_sec;  
int tm\_min;  
int tm\_hour;  
int tm\_mday;  
int tm\_mon;  
int tm\_year;  
int tm\_wday;  
int tm\_yday;  
int tm\_isdst;  
};  
int tm\_sec 代表目前秒数，正常范围为0-59，但允许至61秒  
int tm\_min 代表目前分数，范围0-59  
int tm\_hour 从午夜算起的时数，范围为0-23  
int tm\_mday 目前月份的日数，范围01-31  
int tm\_mon 代表目前月份，从一月算起，范围从0-11  
int tm\_year 从1900 年算起至今的年数  
int tm\_wday 一星期的日数，从星期一算起，范围为0-6  
int tm\_yday 从今年1月1日算起至今的天数，范围为0-365  
int tm\_isdst 日光节约时间的旗标  
此函数返回的时间日期未经时区转换，而是UTC时间。  
   
返回值  返回结构tm代表目前UTC 时间  
   
范例

1. #include <time.h>
2. main(){
3. char \*wday[]={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
4. time\_t timep;
5. struct tm \*p;
6. time(&timep);
7. p=gmtime(&timep);
8. printf(“%d%d%d”,(1900+p->tm\_year), (1+p->tm\_mon),p-
9. >tm\_mday);
10. printf(“%s%d;%d;%d\n”, wday[p->tm\_wday], p->tm\_hour, p
11. ->tm\_min, p->tm\_sec);
12. }

执行  2000/10/28 Sat 8:15:38  
 

localtime（取得当地目前时间和日期）    
相关函数

time, asctime, ctime, gmtime  
   
表头文件

#include  
   
定义函数

struct tm \*localtime(const time\_t \* timep);  
   
函数说明

localtime()将参数timep所指的time\_t结构中的信

息转换成真实世界所使用的时间日期表示方法，然后将结果

由结构tm返回。结构tm的定义请参考gmtime()。此函数返回

的时间日期已经转换成当地时区。  
   
返回值

 返回结构tm代表目前的当地时间。  
   
范例

1. #include<time.h>
2. main(){
3. char \*wday[]={“Sun”,”Mon”,”Tue”,”Wed”,”Thu”,”Fri”,”Sat”};
4. time\_t timep;
5. struct tm \*p;
6. time(&timep);
7. p=localtime(&timep); /\*取得当地时间\*/
8. printf (“%d%d%d ”, (1900+p->tm\_year),( l+p->tm\_mon), p-
9. >tm\_mday);
10. printf(“%s%d:%d:%d\n”, wday[p->tm\_wday],p->tm\_hour, p
11. ->tm\_min, p->tm\_sec);
12. }

执行  2000/10/28 Sat 11:12:22  
 

mktime（将时间结构数据转换成经过的秒数）    
相关函数

time，asctime，gmtime，localtime  
   
表头文件

#include  
   
定义函数

time\_t mktime(strcut tm \* timeptr);  
   
函数说明

mktime()用来将参数timeptr所指的tm结构数据转

换成从公元1970年1月1日0时0分0 秒算起至今的UTC时间

所经过的秒数。  
   
返回值

返回经过的秒数。  
   
范例

/\* 用time()取得时间（秒数），利用localtime()  
转换成struct tm 再利用mktine（）将struct tm转换成原来的

秒数\*/

1. #include<time.h>
2. main()
3. {
4. time\_t timep;
5. strcut tm \*p;
6. time(&timep);
7. printf(“time() : %d \n”,timep);
8. p=localtime(&timep);
9. timep = mktime(p);
10. printf(“time()->localtime()->mktime():%d\n”,timep);
11. }

执行  time():974943297  
time()->localtime()->mktime():974943297  
 

settimeofday（设置目前时间）    
相关函数

time，ctime，ftime，gettimeofday  
   
表头文件

#include  
#include  
   
定义函数

int settimeofday ( const struct timeval \*tv,const

struct timezone \*tz);  
   
函数说明

settimeofday()会把目前时间设成由tv所指的结构

信息，当地时区信息则设成tz所指的结构。详细的说明请参

考gettimeofday()。注意，只有root权限才能使用此函数修改

时间。  
   
返回值

成功则返回0，失败返回－1，错误代码存于errno。  
   
错误代码  EPERM 并非由root权限调用settimeofday（），

权限不够。  
EINVAL 时区或某个数据是不正确的，无法正确设置时间。  
 

time（取得目前的时间）    
相关函数

ctime，ftime，gettimeofday  
   
表头文件

#include  
   
定义函数

time\_t time(time\_t \*t);  
   
函数说明

此函数会返回从公元1970年1月1日的UTC时间从

0时0分0秒算起到现在所经过的秒数。如果t 并非空指针的

话，此函数也会将返回值存到t指针所指的内存。  
   
返回值

成功则返回秒数，失败则返回((time\_t)-1)值，错误

原因存于errno中。  
   
范例

1. #include<time.h>
2. main()
3. {
4. int seconds= time((time\_t\*)NULL);
5. printf(“%d\n”,seconds);
6. }

   
执行  9.73E+08