

程序 14-11 linux/include/time.h

```

1 #ifndef TIME_H
2 #define TIME_H
3
4 #ifndef TIME_T
5 #define TIME_T
6 typedef long time_t; // 从 GMT 1970 年 1 月 1 日午夜 0 时起开始计的时间（秒）。
7 #endif
8
9 #ifndef SIZE_T
10 #define SIZE_T
11 typedef unsigned int size_t;
12 #endif
13
14 #ifndef NULL
15 #define NULL ((void *) 0)
16 #endif
17
18 #define CLOCKS_PER_SEC 100 // 系统时钟滴答频率，100HZ。
19
20 typedef long clock_t; // 从进程开始执行计起的系统经过的时钟滴答数。
21
22 struct tm {
23     int tm_sec; // 秒数 [0, 59]。
24     int tm_min; // 分钟数 [0, 59]。
25     int tm_hour; // 小时数 [0, 59]。
26     int tm_mday; // 1 个月的天数 [0, 31]。
27     int tm_mon; // 1 年中月份 [0, 11]。
28     int tm_year; // 从 1900 年开始的年数。
29     int tm_wday; // 1 星期中的某天 [0, 6] (星期天 =0)。
30     int tm_yday; // 1 年中的某天 [0, 365]。
31     int tm_isdst; // 夏令时标志。正数 - 使用；0 - 没有使用；负数 - 无效。
32 };
33
34 // 判断是否为闰年的宏。
35 #define isleap(year) \
36     ((year) % 4 == 0 && ((year) % 100 != 0 || (year) % 1000 == 0))
37
38 // 以下是有关时间操作的函数原型。
39 // 确定处理器使用时间。返回程序所用处理器时间（滴答数）的近似值。
40 clock_t clock(void);
41 // 取时间（秒数）。返回从 1970.1.1:0:0:0 开始的秒数（称为日历时间）。
42 time_t time(time_t * tp);
43 // 计算时间差。返回时间 time2 与 time1 之间经过的秒数。
44 double difftime(time_t time2, time_t time1);
45 // 将 tm 结构表示的时间转换成日历时间。
46 time_t mktime(struct tm * tp);
47
48 // 将 tm 结构表示的时间转换成字符串。返回指向该串的指针。
49 char * asctime(const struct tm * tp);
50 // 将日历时间转换成字符串形式，如 “Wed Jun 30 21:49:08:1993\n”。
51 char * ctime(const time_t * tp);
52 // 将日历时间转换成 tm 结构表示的 UTC 时间（UTC - 世界时间代码 Universal Time Code）。

```

```
44 struct tm * gmtime(const time\_t *tp);  
    // 将日历时间转换成 tm 结构表示的指定时区 (Time Zone) 的时间。  
45 struct tm *localtime(const time\_t * tp);  
    // 将 tm 结构表示的时间利用格式字符串 fmt 转换成最大长度为 smax 的字符串并将结果存储在 s 中。  
46 size\_t strftime(char * s, size\_t smax, const char * fmt, const struct tm * tp);  
    // 初始化时间转换信息, 使用环境变量 TZ, 对 zname 变量进行初始化。  
    // 在与时区相关的时间转换函数中将自动调用该函数。  
47 void tzset(void);  
48  
49 #endif  
50
```
