

# Turbo C 和 C++ 两类时间和日期函数的使用

江西拖拉机发动机厂 黄焕如

**摘要:** 本文介绍了 Turbo C 和 C++ 的两类时间和日期函数的使用方法, 并列举了两个获取系统时间和日期的实用程序, 可供读者参考。

电脑内部都含有属于其本身的系统时间和日期。早期的微型计算机(如 PC、长城 0520A 等)在开机之前一般都要求用户输入新的时间和日期, 而 286 以上档次的微型计算机都将时间和日期存放在机内的 CMOS 电路中, 能够自动地更新时间和日期。这些设定的时间和日期就是电脑内部系统时间和日期的来源。

Turbo C 和 C++ 有关系统时间和日期的函数主要有两种, 一种定义在 time.h 内, 一种定义在 dos.h 内。

## 一、定义在 time.h 内的 ANSI 标准时间和日期的函数

其中: tm 主要用来存放时间和日期信息

```
struct tm
{
int tm_sec; /* seconds 0--59 */
int tm_min; /* minutes 0--59 */
int tm_hour; /* hours 0--23 */
int tm_mday; /* day of the month, 1--31 */
int tm_mon; /* months since jan., 0--11 */
int tm_year; /* years from 1990 */
int tm_wday; /* days since sunday, 0--6 */
int tm_yday; /* days since jan., 0--356 */
int tm_isdst; /* daylight savings time indicator */
};
```

另外, time\_t (长整数), 也用来存放时间和日期信息, 称为“日历时间”(calendar time)。

这些函数主要包括:

1. time\_t time(time\_t \* time); 以长整数的方式, 存放系统时间和日期信息, 如果需要列出长整数内的时

间和日期, 必须使用相应的转换函数。

2. char \* ctime(time\_t \* time); 将长整数时间和日期, 转换成英文形式: 星期月份 日期 时:分:秒 年。

3. struct tm \* localtime(time\_t \* time); 将长整数存放的时间和日期, 转换成 tm 结构。

4. char \* asctime(struct tm \* ptr); 将 tm 结构转换成英文形式: 星期 月份 日期 时:分:秒 年。

5. struct tm \* gettime(time\_t \* time); 该函数和 localtime() 函数相似, 不同的是前者转换成英车格林威治时间, 后者转换成当地时间和日期的 tm 结构。

6. double difftime(time\_t time2, time\_t time1); 以秒为单位, 计算两个长整数的时间差。

7. clock\_t clock(void); 利用系统时钟计数, 如果需要转换成秒, 则必须除以 CLK\_TCK。

## 二、定义在 dos.h 内的非标准的时间和日期的函数

其中: date 和 time 主要用来存放时间和日期信息:

```
struct date{
int da_year; /* year */
char da_mon; /* month */
char da_day; /* day */
}
struct time{
unsigned char ti_min; /* minutes */
unsigned char ti_hour; /* hour */
unsigned char ti_hund; /* hundredths of seconds */
unsigned char ti_sec; /* seconds */
}
```

这些函数主要包括:

1. void getdate(struct date \* d); 将 DOS 形式的系统日期填入 d 指向的结构 date 中。

2. void gettimeofday (struct time \* t); 将 DOS 形式的系统时间填入 t 指向的结构 time 中。

3. int gettimeofday (int handle, struct ftime \* fptr); 将 handle 所联结的文件返回该文件的生成时间和日期, 信息放入 fptr 指向的结构。

4. void setdate (struct date \* d); 将 d 指向的结构中的指定值设置在 DOS 系统日期中。

5. void settime (struct time \* t); 将 t 指向的结构中的指定值设置在 DOS 系统时间中。

6. void unixtodostime (time\_t utime, struct date \* d, struct time \* t); 将 UNIX 系统时间格式转换成 DOS 系统格式, utime 保存 UNIX 时间格式, 对应的 DOS 系统的日期和时间被保存在由 d 和 t 指向的结构中。

### 三、系统时间和日期函数的作用示例

以下示例分别为系统时间和日期的调用程序, 程序在屏幕选定位置画出一时钟模型, 巧妙地利用 SIN() 和 COS() 函数确定时针、分针、秒针的坐标位置, 构画出时钟的时、分、秒针图。该程序稍加修改, 就能演变成不断变化的时种。程序在屏幕选定位置画出当前年月的图画, 当天的日期使用红色特殊地表示。该程序稍加修改, 就能输出任何年份的“年历”图。

/\* 程序 1 系统时间示例 \*/

```
#include "stdio.h"
#include "stdlib.h"
#include "alloc.h"
#include "graphics.h"
#include "fcntl.h"
#include "fcntl.h"
#include "dos.h"
#include "math.h"
Time (int x1, int y1)
{
char stime [20];
struct time * dos_time;
int x0[12], y0[12], i; char * p;
float x, y, dt1, dt2, dt3;

setfillstyle(SOLID_FILL, LIGHTGRAY);
bar3d (x1, y1, x1+300, y1+160, 0, 0);
bar3d (x1+1, y1+1, x1+200-1, y1+160, 0, 0);
x1 = x1+100; y1 = y1+80;
setfillstyle(SOLID_FILL, LIGHTCYAN);
setcolor(LIGHTCYAN);
pieslice(x1, y1, 0, 360, 70);
setcolor(LIGHTRED);
```

```
settextstyle(VERT_DIR, 0, 1);
x0[0] = x1+30; y0[0] = y1-60;
x0[1] = x1+45; y0[1] = y1-35;
x0[2] = x1+55; y0[2] = y1-8;
x0[3] = x1+46; y0[3] = y1+20;
x0[4] = x1+26; y0[4] = y1+40;
x0[5] = x1-6; y0[5] = y1+50;
x0[6] = x1-35; y0[6] = y1+40;
x0[7] = x1-55; y0[7] = y1+20;
x0[8] = x1-65; y0[8] = y1-8;
y0[9] = x1-60; y0[9] = y1-35;
x0[10] = x1-40; y0[10] = y1-60;
x0[11] = x1-10; y0[11] = y1-70;
for (i = 0; i <= 11; i++){
itoa(i+1, p, 10);
outtextxy (x0[i], y0[i], p);
}
gettime (dos_time);
dt1 = dos_time->ti_hour;
dt2 = dos_time->ti_min;
dt2 = dos_time->ti_min;
dt3 = dos_time->ti_sec;
sprintf (stime, "time is %d : %d : %d",
dos_time->ti_hour, dos_time->ti_min,
dos_time->ti_sec);
```

```
setcolor(YELLOW);
outtextxy (x1-100+30, y1-80-30, stime);
setcolor(WHITE);
x = 55 * cos (30 * (dt3 / 5 + 9) * 3.14159 / 180);
y = 55 * sin (30 * (dt3 / 5 + 9) * 3.14159 / 180);
moveto(x1, y1); linerel (x, y);
```

```
setcolor(BROWN);
x = 43 * COS(30 * (dt2 / 5 + 9 +
dt3 / 5 + 9) / 60) * 3.14159 / 180);
y = 43 * sin (30 * (dt2 / 5 + 9 +
(dt3 / 5 + 9) / 60) * 3.14159 / 180);
moveto(x1, y1); linerel (x, y);
x = 35 * cos (30 * (dt1 + 9 +
(dt3 / 5 + 9) / 60) / 60) * 3.14159 / 180);
y = 35 * sin (30 * (dt1 + 9 + (dt2 / 5 + 9 +
(dt3 / 5 + 9) / 60) / 60) * 3.14159 / 180);
moveto(x1, y1); linerel (x, y);
moveto(x1+1, y1+1); linerel (x, y);
getch ();
}
main()
{
int driver = VGA, mode = 2;
registerbgidriver (EGAVGA_driver);
initgraph (&driver, &mode)
Time (50, 50);
```

```

closegraph();
}
/* 程序 Z 系统日期示例 */
#include "stdio.h"
#include "time.h"
#include "conio.h"
#include "graphics.h"
#include "stdlib.h"
#define X 50
#define Y 50

int year, uruu, year__days, week;
int month, dt1, dt2, dt3;
int mdays [12] = {31, 28, 31, 30, 31, 30, 31, 30, 31, 30, 31, 30};
char sdate [20];

main()
{
int i; char * p;
int driver = VGA, mode = 2;
stuvct date * dos __date;
registerbgidriver (EGAVGA__driver);
initgraph (&driver, &mode, "");

setfillstyle(SOLID__FILL, LIGHTGRAY);
bar3d (X, Y, X+260, Y+200, 0, 0);
setfillstyle(SOLID__FILL < LIGHTCYAN);
setcolor(LIGHTRED);
setttextstyle(VERT__DIR, 0, 1);

getdate (dos __date); /* 取系统日期 */
dt1 = dos __date->da__year;
dt2 = dos __date->da__mon;
dt3 = dos __date->da__day;
sprintf (sdate, "Doday is %d.%d.%d".
dos __date->da__year,
dos __date->da__mon, dos __date->da__day);
year = dos __date->da__year;
month = dos __date->da__mon;
year__check ();
week__check ();

outtextxy (X+50, Y-30, sdate);
setcolor(BROWN);
setttextstyle(0, HORIZ__DIR, 1);
outtextxy (X+20, Y+20,
"SUN MON TUE WED THO FRT SAT");
setcolor(BLACK);
for (i = 1; i < 13; i++){if(dt2 == i)
week = xs(i, week);
else
week = xs0(i, week);}
getch ();
closegraph();
}

year__check () /* 处理闰年情况 */
{
if(year-4 * (year / 4)) == 0) uruu = 1;
if((year-100 * (year / 100 == 0)) uruu = 0;
if((year-100 * (year / 400)) == 0) uruu = 1;
mdays [1] += uruu;
}

week__check () /* 计算机年首日是星期几 */
{
int i;
i = year+year / 4 -year / 100 +year / 400; week = i -7 * (i / 7);
if(uruu == 1) week -= 1;
}

xs(month, week)
int month, week;
{
int k = 20, j = 0, i = 0; char * p;
while (i < week){j = (i + 1) * 30; ++i;}
for (i = 1; i <= mdays [month - 1]; i++)
{itoa(i, p, 10;
if(dt3 == i)
setcolor(RED);
else
setcolor(BLACK);
outtextxy (x + 30 + j, y + 20 + k, p);
if(((i + week) - 7 * ((i + week) / 7)) == 0)
{j = 0; k = k + 25; }else j'j + 30;
}
}

xs0 (month, week)
int month, week;
{
int i = 0;
while (i < week) ++i;
for (i + 1; i <= mdays [month - 1]; i++){
return((week + mdays [month - 1] - 7 *
((week + mdays [month - 1] / 7))););
}
}

```

▲