

```
void DateTime::splitTicks(time_t time) {
    seconds = time % 60;
    time /= 60;
    minutes = time % 60;
    time /= 60;
    hours = time % 24;
    time /= 24;

    year = DateTime::reduceDaysToYear(time);
    month = DateTime::reduceDaysToMonths(time,year);
    day = int(time);
}

int DateTime::reduceDaysToYear(time_t &days) {
    int year;
    for (year=1970;days>daysInYear(year);year++) {
        days -= daysInYear(year);
    }
    return year;
}

int DateTime::reduceDaysToMonths(time_t &days,int year) {
    int month;
    for (month=0;days>daysInMonth(month,year);month++)
        days -= daysInMonth(month,year);
    return month;
}
```

```

*
* gmtime - convert the calendar time into broken down time
*/
/* $Header: gmtime.c,v 1.4 91/04/22 13:20:27 ceriel Exp $ */

#include      <time.h>
#include      <limits.h>
#include      "loc_time.h"

struct tm *
gmtime(register const time_t *timer)
{
    static struct tm br_time;
    register struct tm *timep = &br_time;
    time_t time = *timer;
    register unsigned long dayclock, dayno;
    int year = EPOCH_YR;

    dayclock = (unsigned long)time % SECS_DAY;
    dayno = (unsigned long)time / SECS_DAY;

    timep->tm_sec = dayclock % 60;
    timep->tm_min = (dayclock % 3600) / 60;
    timep->tm_hour = dayclock / 3600;
    timep->tm_wday = (dayno + 4) % 7;          /* day 0 was a thursday */
    while (dayno >= YEARSIZE(year)) {
        dayno -= YEARSIZE(year);
        year++;
    }
    timep->tm_year = year - YEAR0;
    timep->tm_yday = dayno;
    timep->tm_mon = 0;
    while (dayno >= _ytab[LEAPYEAR(year)][timep->tm_mon]) {
        dayno -= _ytab[LEAPYEAR(year)][timep->tm_mon];
        timep->tm_mon++;
    }
    timep->tm_mday = dayno + 1;
    timep->tm_isdst = 0;

    return timep;
}

```

```

#define EPOCH_YR 1970
#define LEAPYEAR(year) (!((year) % 4) && (((year) % 100) || !((year) % 400)))
#define YEARSIZE(year) (LEAPYEAR(year) ? 366 : 365)

int year = EPOCH_YR;

while (dayno >= YEARSIZE(year)) {
    dayno -= YEARSIZE(year);
    year++;
}

```

### ISO 9601 Epoch (Year 0 = 1 BC)

```

#define CYCLE_1    365
#define CYCLE_4    (CYCLE_1 * 4 + 1)
#define CYCLE_100  (CYCLE_4 * 25 - 1)
#define CYCLE_400  (CYCLE_100 * 4 + 1)

year += 400 * (dayno / CYCLE_400)
dayno = dayno % CYCLE_400

year += 100 * (dayno / CYCLE_100)
dayno = dayno % CYCLE_100

year += 4 * (dayno / CYCLE_4)
dayno = dayno % CYCLE_4

year += 1 * (dayno / CYCLE_1)
dayno = dayno % CYCLE_1

```