

# C gmtime

[Programming C]

## gmtime

time 関数呼び出して、低水準の時刻の値を取得し、gmtime を利用して日付と時刻の表現に変換する。

```
#include <time.h>
struct tm *gmtime(const time_t timeval);
```

member	説明
int tm_sec	秒 (0-61)
int tm_min	分 (0-59)
int tm_hour	時間 (0-23)
int tm_mday	日 (1-31)
int tm_mon	月 (0-11)
int tm_year	年 (1900 年が起点)
int tm_wday	曜日 (0-6) 0 は月曜日
int tm_yday	1 年の中での日数 (0-365)
int tm_isdst	夏時間

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    struct tm *tm_ptr;
    time_t the_time;

    (void) time(&the_time);
    tm_ptr = gmtime(&the_time);

    printf("%02d/%02d/%02d %02d:%02d:%02d\n",
           tm_ptr->tm_year + 1900,
           tm_ptr->tm_mon + 1,
           tm_ptr->tm_mday,
           tm_ptr->tm_hour,
           tm_ptr->tm_min,
           tm_ptr->tm_sec);
    exit(0);
}
```

## localtime

```
#include <time.h>
struct tm *localtime(const time_t *timeval);
```

- local time zone と夏時間に関する調整を加えた値を返す。
- 使い方は、gmtime と同様

## mktime

```
#include <time.h>
```

```
time_t mktime(struct tm *timeptr);
```

- tm 構造体に格納した値から逆に time\_t 型の値を求める。

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
int main()
{
    struct tm *tm_ptr;
    time_t the_time;
    time_t other_time;

    (void) time(&the_time);
    tm_ptr = localtime(&the_time);

    printf("%d\n", the_time);
    printf("%02d/%02d/%02d %02d:%02d:%02d\n",
           tm_ptr->tm_year + 1900,
           tm_ptr->tm_mon + 1,
           tm_ptr->tm_mday,
           tm_ptr->tm_hour,
           tm_ptr->tm_min,
           tm_ptr->tm_sec);

    other_time = mktime(tm_ptr);
    printf("%d\n", other_time);
    exit(0);
}
```

## asctime、ctime

- asctime 関数は、tm 構造体 timeptr で指定された日付と時刻を表す文字列を返す。
- ctime 関数は、asctime(localtime(timeval)) と同じ。

```
#include <time.h>
char *asctime(const struct tm *timeptr);
char *ctime(const time_t *timeval);
```

返される文字は次のような形式

```
Sun Oct 14 01:46:51 2007\n#0
```

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    struct tm *tm_ptr;
    time_t the_time;

    (void) time(&the_time);
    tm_ptr = localtime(&the_time);

    printf("%s", asctime(tm_ptr));
    printf("%s", ctime(&the_time)); exit(0);
}
```

## strftime

- timeptr で指定した構造体 tm が表す日付と時刻に書式を設定する。

```
#include <time.h>
```

```
size_t strftime(char *s, size_t maxsize, const char *format, struct tm *timeptr);
```

変換指定子	説明
%a	曜日の省略
%A	曜日
%b	月名の省略
%B	月名
%c	日付と時刻
%d	日
%H	24 時間制の時間
%I	12 時間制の時間
%m	月
%M	分
%p	AM または PM
%S	秒
%Y	西暦年

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    struct tm *tm_ptr;
    time_t the_time;
    char buf[256];
    (void) time(&the_time);
    tm_ptr = localtime(&the_time);
    strftime(buf, 256, "%Y/%m/%d %A %H:%M:%S", tm_ptr);
    printf("%s\n", buf);    exit(0);
}
```

2007/10/15 Monday 22:50:48

## strftime

日付を読み取る際には、strftime を使う。

日付と時刻を表す文字列を受け取って、tm 構造体を作成

```
#include <time.h>
char *strftime(const char *buf, const char *format, struct tm *timeptr);
```

---

この本からの覚書。