

Б10. Функции, обработващи датата и времето: <time.h>

В заглавния файл <time.h> са декларирани типове и функции, които обработват дати и часове. Някои функции обработват местното време; ето защо то може да се различава от календарното време, поради различия в часовата зона например. `clock_t` и `time_t` са аритметичните типове, които представят времето; в `struct tm` се съдържат компонентите на календарното време.

```
int tm sec;      секунди (0,61)
int tm min;      минути (0,59)
int tm hour;     часове след полунощ (0,23)
int tm mday;     ден от месеца (1,31)
int tm mon;      месеци, като броенето започва от януари
                (0,11)
int tm year;     години след 1900
int tm wday;     дни след неделя (0,6)
int tm yday;     дни след 1 януари (0,365)

int tm isdst;    флаг за лятното часово време
```

`tm_isdst` има положителна стойност по време на лятното часово време, нула извън него и отрицателна стойност, ако няма налична информация.

```
clock_t clock(void)
```

`clock` връща времето на процесора, използвано от програмата от началото на нейното изпълнение; ако информацията не е налична, `clock` връща -1. `clock () /CLOCKS_PER_SEC` дава времето в секунди.

```
time_t time(time_t *tp)
```

`time` връща текущото календарно време или -1, ако такова не е налично. Ако `tp` не е `NULL`, върнатата стойност се присвоява на `*tp`.

```
double difftime(time_t time2, time_t time1)
```

`difftime` връща разликата от часовете `time2-time1` в секунди.

```
time_t mktime(struct tm *tp)
```

`mktime` преобразува локалното време от структурата `*tp` в календарно време, като запазва представянето, което `time` използва. Компонентите на времето притежават стойности в горепосочените области, `mktime` връща календарното време или -1, ако то не може да бъде представено в съответната форма.

Следващите четири функции връщат указатели към статични обекти, върху които може да се записва информация от следващи извиквания.

```
char *asctime(const struct tm *tp)
```

`asctime` преобразува времето в структурата `*tp` до низ от следния вид:

```
Sun Jan 3 15:14:13 1988\n\0
```

```
char *ctime(const time_t *tp)
```

`ctime` преобразува календарното време `*tp` в локално; функцията е еквивалентна на

```
asctime(localtime(tp))
```

`struct tm *gmtime(const time_t *tp)`
gmtime преобразува календарното време *tp в UTC (Coordinate Universal Time). Ако няма достъп до времето по UTC, функцията връща NULL.

`struct tm *localtime(const time_t *tp)`
localtime преобразува календарното време *tp в локално.

`size_t strftime(char *s, size_t smax, const char *fmt, const struct tm *tp)`
strftime форматира информацията за датата и времето от *tp в s, в съответствие с f tm (която е аналог на форматирането с printf). Всички обикновени символи (включително '\0') се копират в s. Всяка последователност от символи от вида %c се замества, както е показано по-долу, като се използват подходящи за локалната среда стойности. В s не могат да се поставят повече от smax на брой символа, strftime връща броя на символите без '\0' или нула, ако е имало повече от smax символа.

%a	абривиатура на име на ден от седмицата.
%A	пълно име на ден от седмицата.
%b	абривиатура на име на месец.
%B	пълно име на месец.
%c	локално представяне на датата и часа.
%d	ден от месеца (01-31).
%H	час (24-часов часовник) (00-23).
%I	час (12-часов часовник) (01-12).
%j	ден от годината (001-366).
%m	месец (01-12).
%M	минута (00-59).
%p	локален еквивалент на AM или PM.
%S	секунди (00-61).
%U	пореден номер на седмицата от годината (неделя се счита за първи ден от седмицата) (00-53).
%w	ден от седмицата (0-6, неделя е 0).
%W	пореден номер на седмицата от годината (понеделник се счита за първи ден от седмицата) (00-53).
%x	локално представяне на датата.
%X	локално представяне на часа.
%y	година без века, в който участва (00-99).
%Y	година с века, в който участва.
%Z	име на времевата зона, ако такава съществува.
%%	%