## NAME

ctime, localtime, gmtime, asctime, tzset − convert date and time to ASCII

## SYNOPSIS

**#include <sys/types.h>**
**#include <time.h>**

**void tzset(void)**
**char *ctime(const time_t ***clock***)**
**char *asctime(const struct tm ***tm***)**
**struct tm *localtime(const time_t ***clock***)**
**struct tm *gmtime(const time_t ***clock***)**

## DESCRIPTION

**Tzset** uses the value of the environment variable **TZ** to set up the time conversion information used by **localtime**.

If **TZ** does not appear in the environment, the **TZDEFAULT** file (as defined in <*tzfile.h*>) is used by **localtime**. If this file fails for any reason, the GMT offset as provided by the kernel is used. In this case, DST is ignored, resulting in the time being incorrect by some amount if DST is currently in effect. If this fails for any reason, GMT is used.

If **TZ** appears in the environment but its value is a null string, Greenwich Mean Time is used; if **TZ** appears and begins with a slash, it is used as the absolute pathname of the **tzfile**(5)-format file from which to read the time conversion information; if **TZ** appears and begins with a character other than a slash, it's used as a pathname relative to the system time conversion information directory, defined as **TZDIR** in the include file **tzfile.h**. If this file fails for any reason, the GMT offset as provided by the kernel is used, as described above. If this fails for any reason, GMT is used. See **TZ**(5) for a proper description of the **TZ** variable.

**Ctime** converts a time value, pointed to by *clock*, such as returned by **time**(2) into ASCII and returns a pointer to a 26-character string in the following form. All the fields have constant width.

```
Sun Sep 16 01:03:52 1973\n\0
```

**Localtime** and **gmtime** return pointers to structures containing the broken-down time. **Localtime** corrects for the time zone and possible daylight savings time; **gmtime** converts directly to GMT, which is the time UNIX uses. **Asctime** converts a broken-down time to ASCII and returns a pointer to a 26-character string.

The structure declaration from the include file is:

```
struct tm {
        int tm_sec;       /* 0-59    seconds */
        int tm_min;       /* 0-59    minutes */
        int tm_hour;      /* 0-23    hour */
        int tm_mday;      /* 1-31    day of month */
        int tm_mon;       /* 0-11    month */
        int tm_year;      /* 0-      year − 1900 */
        int tm_wday;      /* 0-6     day of week (Sunday = 0) */
        int tm_yday;      /* 0-365   day of year */
        int tm_isdst;     /* flag:   daylight savings time in effect */
        long tm_gmtoff;   /* offset from GMT in seconds */
        char **tm_zone;   /* abbreviation of timezone name */
};
```

**Tm_isdst** is non-zero if a time zone adjustment such as Daylight Savings time is in effect.

**Tm_gmtoff** is the offset (in seconds) of the time represented from GMT, with positive values indicating East of Greenwich.

## FILES

| | |
|---|---|
| /usr/lib/zoneinfo | time zone information directory |
| /etc/localtime | local time zone file |

## SEE ALSO

**time**(2), **getenv**(3), **tzfile**(5), **TZ**(5), **environ**(7), **zic**(8).

## NOTE

The return values point to static data whose content is overwritten by each call. The **tm_zone** field of a returned **struct tm** points to a static array of characters, which will also be overwritten at the next call (and by calls to **tzset**).