NAME

tzset, tzname, timezone, daylight − initialize time conversion information

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

#include <time.h>

void tzset(void);

extern char ****tzname*[2];**
extern long *timezone***;**
extern int *daylight***;**

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

tzset():
_POSIX_C_SOURCE

tzname:
_POSIX_C_SOURCE

timezone, *daylight*:
_XOPEN_SOURCE
|| /* glibc >= 2.19: */ _DEFAULT_SOURCE
|| /* glibc <= 2.19: */ _SVID_SOURCE

DESCRIPTION

The **tzset**() function initializes the *tzname* variable from the **TZ** environment variable. This function is automatically called by the other time conversion functions that depend on the timezone. In a System-V-like environment, it will also set the variables *timezone* (seconds West of UTC) and *daylight* (to 0 if this timezone does not have any daylight saving time rules, or to nonzero if there is a time, past, present, or future when daylight saving time applies).

If the **TZ** variable does not appear in the environment, the system timezone is used. The system timezone is configured by copying, or linking, a file in the **tzfile**(5) format to */etc/localtime*. A timezone database of these files may be located in the system timezone directory (see the **FILES** section below).

If the **TZ** variable does appear in the environment, but its value is empty, or its value cannot be interpreted using any of the formats specified below, then Coordinated Universal Time (UTC) is used.

The value of **TZ** can be one of two formats. The first format is a string of characters that directly represent the timezone to be used:

```
std offset[dst[offset][,start[/time],end[/time]]]
```

There are no spaces in the specification. The *std* string specifies an abbreviation for the timezone and must be three or more alphabetic characters. When enclosed between the less-than (<) and greater-than (>) signs, the character set is expanded to include the plus (+) sign, the minus (−) sign, and digits. The *offset* string immediately follows *std* and specifies the time value to be added to the local time to get Coordinated Universal Time (UTC). The *offset* is positive if the local timezone is west of the Prime Meridian and negative if it is east. The hour must be between 0 and 24, and the minutes and seconds 00 and 59:

```
[+|−]hh[:mm[:ss]]
```

The *dst* string and *offset* specify the name and offset for the corresponding daylight saving timezone. If the offset is omitted, it defaults to one hour ahead of standard time.

The *start* field specifies when daylight saving time goes into effect and the *end* field specifies when the change is made back to standard time. These fields may have the following formats:

J*n* This specifies the Julian day with *n* between 1 and 365. Leap days are not counted. In this format, February 29 can't be represented; February 28 is day 59, and March 1 is always day 60.

n This specifies the zero-based Julian day with *n* between 0 and 365. February 29 is counted in leap years.

M*m*.*w*.*d*

This specifies day *d* (0 <= *d* <= 6) of week *w* (1 <= *w* <= 5) of month *m* (1 <= *m* <= 12). Week 1 is the first week in which day *d* occurs and week 5 is the last week in which day *d* occurs. Day 0 is a Sunday.

The *time* fields specify when, in the local time currently in effect, the change to the other time occurs. If omitted, the default is 02:00:00.

Here is an example for New Zealand, where the standard time (NZST) is 12 hours ahead of UTC, and daylight saving time (NZDT), 13 hours ahead of UTC, runs from the first Sunday in October to the third Sunday in March, and the changeovers happen at the default time of 02:00:00:

```
TZ="NZST-12:00:00NZDT-13:00:00,M10.1.0,M3.3.0"
```

The second format specifies that the timezone information should be read from a file:

```
:[filespec]
```

If the file specification *filespec* is omitted, or its value cannot be interpreted, then Coordinated Universal Time (UTC) is used. If *filespec* is given, it specifies another **tzfile**(5)-format file to read the timezone information from. If *filespec* does not begin with a '/', the file specification is relative to the system timezone directory. If the colon is omitted each of the above **TZ** formats will be tried.

Here's an example, once more for New Zealand:

```
TZ=":Pacific/Auckland"
```

ENVIRONMENT

TZ If this variable is set its value takes precedence over the system configured timezone.

TZDIR

If this variable is set its value takes precedence over the system configured timezone database directory path.

FILES

/etc/localtime

The system timezone file.

/usr/share/zoneinfo/

The system timezone database directory.

/usr/share/zoneinfo/posixrules

When a TZ string includes a dst timezone without anything following it, then this file is used for the start/end rules. It is in the **tzfile**(5) format. By default, the zoneinfo Makefile hard links it to the *America/New_York* tzfile.

Above are the current standard file locations, but they are configurable when glibc is compiled.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes**(7).

Interface	Attribute	Value
tzset()	Thread safety	MT-Safe env locale

STANDARDS

POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD.

NOTES

4.3BSD had a function **char *timezone(***zone***,** *dst***)** that returned the name of the timezone corresponding to its first argument (minutes West of UTC). If the second argument was 0, the standard name was used, otherwise the daylight saving time version.

SEE ALSO

date(1), **gettimeofday**(2), **time**(2), **ctime**(3), **getenv**(3), **tzfile**(5)