NAME

gettimeofday, **settimeofday** - get/set date and time

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/time.h>

int
gettimeofday(*struct timeval *tp, struct timezone *tzp*);

int
settimeofday(*const struct timeval *tp, const struct timezone *tzp*);

DESCRIPTION

The system's notion of the current Greenwich time and the current time zone is obtained with the **gettimeofday**() system call, and set with the **settimeofday**() system call. The time is expressed in seconds and microseconds since midnight (0 hour), January 1, 1970. The resolution of the system clock is hardware dependent, and the time may be updated continuously or in "ticks". If *tp* or *tzp* is NULL, the associated time information will not be returned or set.

The structures pointed to by *tp* and *tzp* are defined in <*sys/time.h*> as:

```
struct timeval {
        time_t          tv_sec;         /* seconds */
        suseconds_t     tv_usec;   /* and microseconds */
};

struct timezone {
        int     tz_minuteswest; /* minutes west of Greenwich */
        int     tz_dsttime;     /* type of dst correction */
};
```

The *timezone* structure indicates the local time zone (measured in minutes of time westward from Greenwich), and a flag that, if nonzero, indicates that Daylight Saving time applies locally during the appropriate part of the year. The kernel generally does not track these values and they are usually returned as zero. Use localtime(3) to find the offset for the currently active timezone.

Only the super-user may set the time of day or time zone. If the system is running at securelevel >= 2

(see init(8)), the time may only be advanced or retarded by a maximum of one second. This limitation is imposed to prevent a malicious super-user from setting arbitrary time stamps on files. The system time can be adjusted backwards without restriction using the adjtime(2) system call even when the system is secure.

RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The following error codes may be set in *errno*:

[EINVAL] The supplied *timeval* value is invalid.

[EPERM] A user other than the super-user attempted to set the time.

SEE ALSO

date(1), adjtime(2), clock_gettime(2), ctime(3), timeradd(3), clocks(7)

HISTORY

The **gettimeofday**() system call appeared in 4.2BSD.