

**NAME**

`adjtime` - correct the time to synchronize the system clock

**SYNOPSIS**

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

```
int adjtime(const struct timeval *delta, struct timeval *olddelta);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

```
adjtime(): _BSD_SOURCE
```

**DESCRIPTION**

The `adjtime()` function gradually adjusts the system clock (as returned by [gettimeofday\(2\)](#)). The amount of time by which the clock is to be adjusted is specified in the structure pointed to by *delta*. This structure has the following form:

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

If the adjustment in *delta* is positive, then the system clock is speeded up by some small percentage (i.e., by adding a small amount of time to the clock value in each second) until the adjustment has been completed. If the adjustment in *delta* is negative, then the clock is slowed down in a similar fashion.

If a clock adjustment from an earlier `adjtime()` call is already in progress at the time of a later `adjtime()` call, and *delta* is not NULL for the later call, then the earlier adjustment is stopped, but any already completed part of that adjustment is not undone.

If *olddelta* is not NULL, then the buffer that it points to is used to return the amount of time remaining from any previous adjustment that has not yet been completed.

**RETURN VALUE**

On success, `adjtime()` returns 0. On failure, -1 is returned, and *errno* is set to indicate the error.

**ERRORS****EINVAL**

The adjustment in *delta* is outside the permitted range.

**EPERM**

The caller does not have sufficient privilege to adjust the time. Under Linux, the `CAP_SYS_TIME` capability is required.

**ATTRIBUTES****Multithreading (see [pthreads\(7\)](#))**

The `adjtime()` function is thread-safe.

**CONFORMING TO**

4.3BSD, System V.

**NOTES**

The adjustment that `adjtime()` makes to the clock is carried out in such a manner that the clock is always monotonically increasing. Using `adjtime()` to adjust the time prevents the problems that can be caused for certain applications (e.g., [make\(1\)](#)) by abrupt positive or negative jumps in the system time.

`adjtime()` is intended to be used to make small adjustments to the system time. Most systems impose a limit on the adjustment that can be specified in *delta*. In the glibc implementation, *delta* must be less than or equal to  $(\text{INT\_MAX} / 1000000 - 2)$  and greater than or equal to  $(\text{INT\_MIN} / 1000000 + 2)$  (respectively 2145 and -2145 seconds on i386).

**BUGS**

A longstanding bug meant that if *delta* was specified as NULL, no valid information about the outstanding clock adjustment was returned in *olddelta*. (In this circumstance, `adjtime()` should return the outstanding

clock adjustment, without changing it.) This bug is fixed on systems with glibc 2.8 or later and Linux kernel 2.6.26 or later.

**SEE ALSO**

[adjtimex\(2\)](#), [gettimeofday\(2\)](#), [time\(7\)](#)

**COLOPHON**

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.