

**NAME**

adjtimex - tune kernel clock

**SYNOPSIS**

```
#define _BSD_SOURCE /* See feature_test_macros(7)
*/"
#include <sys/timex.h>

int adjtimex(struct timex *buf);
```

**DESCRIPTION**

Linux uses David L. Mills' clock adjustment algorithm (see RFC 1305). The system call **adjtimex()** reads and optionally sets adjustment parameters for this algorithm. It takes a pointer to a *timex* structure, updates kernel parameters from field values, and returns the same structure with current kernel values. This structure is declared as follows:

```
struct timex {
    int modes; /* mode selector */
    long offset; /* time offset (usec) */
    long freq; /* frequency offset (scaled ppm) */
    long maxerror; /* maximum error (usec) */
    long esterror; /* estimated error (usec) */
    int status; /* clock command/status */
    long constant; /* pll time constant */
    long precision; /* clock precision (usec) (read-only) */
    long tolerance; /* clock frequency tolerance (ppm)
(read-only) */
    struct timeval time; /* current time (read-only) */
    long tick; /* usecs between clock ticks */
};
```

The *modes* field determines which parameters, if any, to set. It may contain a bitwise-*or* combination of zero or more of the following bits:

```
#define ADJ_OFFSET 0x0001 /* time offset */
#define ADJ_FREQUENCY 0x0002 /* frequency offset */
#define ADJ_MAXERROR 0x0004 /* maximum time error */
#define ADJ_ESTERROR 0x0008 /* estimated time error */
#define ADJ_STATUS 0x0010 /* clock status */
#define ADJ_TIMECONST 0x0020 /* pll time constant */
#define ADJ_TICK 0x4000 /* tick value */
#define ADJ_OFFSET_SINGLESHOT 0x8001 /* old-fashioned adjtime() */
```

Ordinary users are restricted to a zero value for *modes*. Only the superuser may set any parameters.

**RETURN VALUE**

On success, **adjtimex()** returns the clock state:

```
#define TIME_OK 0 /* clock synchronized */
#define TIME_INS 1 /* insert leap second */
#define TIME_DEL 2 /* delete leap second */
#define TIME_OOP 3 /* leap second in progress */
#define TIME_WAIT 4 /* leap second has occurred */
#define TIME_BAD 5 /* clock not synchronized */
```

On failure, **adjtimex()** returns -1 and sets *errno*.

**ERRORS****EFAULT**

*buf* does not point to writable memory.

**EINVAL**

An attempt is made to set *buf.offset* to a value outside the range -131071 to +131071, or to set *buf.status* to a value other than those listed above, or to set *buf.tick* to a value outside the range 900000/HZ to 1100000/HZ, where **HZ** is the system timer interrupt frequency.

**EPERM**

*buf.modes* is nonzero and the caller does not have sufficient privilege. Under Linux the **CAP\_SYS\_TIME** capability is required.

**CONFORMING TO**

**adjtimex()** is Linux-specific and should not be used in programs intended to be portable. See [adjtime\(3\)](#) for a more portable, but less flexible, method of adjusting the system clock.

**SEE ALSO**

[settimeofday\(2\)](#), [adjtime\(3\)](#), [capabilities\(7\)](#), [time\(7\)](#), [adjtimex\(8\)](#)

**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/>.