



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'timerisset.3'

\$ man timerisset.3

TIMERADD(3) Linux Programmer's Manual TIMERADD(3)

NAME

timeradd, timersub, timercmp, timerclear, timerisset - timeval operations

SYNOPSIS

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

void timeradd(struct timeval *a, struct timeval *b,
              struct timeval *res);

void timersub(struct timeval *a, struct timeval *b,
              struct timeval *res);

void timerclear(struct timeval *tvp);

int timerisset(struct timeval *tvp);

int timercmp(struct timeval *a, struct timeval *b, CMP);
```

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

All functions shown above:

Since glibc 2.19:

 _DEFAULT_SOURCE

Glibc 2.19 and earlier:

 _BSD_SOURCE

DESCRIPTION

The macros are provided to operate on timeval structures, defined in <sys/time.h> as:

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

```
};
```

`timeradd()` adds the time values in `a` and `b`, and places the sum in the `timeval` pointed to by `res`. The result is normalized such that `res->tv_usec` has a value in the range 0 to 999,999.

`timersub()` subtracts the time value in `b` from the time value in `a`, and places the result in the `timeval` pointed to by `res`. The result is normalized such that `res->tv_usec` has a value in the range 0 to 999,999.

`timerclear()` zeros out the `timeval` structure pointed to by `tv`, so that it represents the Epoch: 1970-01-01 00:00:00 +0000 (UTC).

`timerisset()` returns true (nonzero) if either field of the `timeval` structure pointed to by `tv` contains a nonzero value.

`timercmp()` compares the timer values in `a` and `b` using the comparison operator `CMP`, and returns true (nonzero) or false (0) depending on the result of the comparison. Some systems (but not Linux/glibc), have a broken `timercmp()` implementation, in which `CMP` of `>=`, `<=`, and `==` do not work; portable applications can instead use

```
!timercmp(..., <)
```

```
!timercmp(..., >)
```

```
!timercmp(..., !=)
```

RETURN VALUE

`timerisset()` and `timercmp()` return true (nonzero) or false (0).

ERRORS

No errors are defined.

CONFORMING TO

Not in POSIX.1. Present on most BSD derivatives.

SEE ALSO

`gettimeofday(2)`, `time(7)`

COLOPHON

This page is part of release 5.10 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 <https://www.kernel.org/doc/man-pages/>.