



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'pthread\_timedjoin\_np.3'***

***\$ man pthread\_timedjoin\_np.3***

PTHREAD\_TRYJOIN\_NP(3) Linux Programmer's Manual PTHREAD\_TRYJOIN\_NP(3)

NAME

pthread\_tryjoin\_np, pthread\_timedjoin\_np - try to join with a terminated thread

SYNOPSIS

```
#define _GNU_SOURCE /* See feature_test_macros(7) */
#include <pthread.h>
int pthread_tryjoin_np(pthread_t thread, void **retval);
int pthread_timedjoin_np(pthread_t thread, void **retval,
    const struct timespec *abstime);
```

Compile and link with -pthread.

DESCRIPTION

These functions operate in the same way as pthread\_join(3), except for the differences described on this page.

The pthread\_tryjoin\_np() function performs a nonblocking join with the thread thread, returning the exit status of the thread in \*retval. If thread has not yet terminated, then instead of blocking, as is done by pthread\_join(3), the call returns an error.

The `pthread_timedjoin_np()` function performs a join-with-timeout. If thread has not yet terminated, then the call blocks until a maximum time, specified in `abstime`, measured against the `CLOCK_REALTIME` clock. If the timeout expires before thread terminates, the call returns an error. The `abstime` argument is a structure of the following form, specifying an absolute time measured since the Epoch (see `time(2)`):

```
struct timespec {
    time_t tv_sec; /* seconds */
    long tv_nsec; /* nanoseconds */
};
```

### RETURN VALUE

On success, these functions return 0; on error, they return an error number.

### ERRORS

These functions can fail with the same errors as `pthread_join(3)`.

`pthread_tryjoin_np()` can in addition fail with the following error:

`EBUSY` thread had not yet terminated at the time of the call.

`pthread_timedjoin_np()` can in addition fail with the following errors:

#### ETIMEDOUT

The call timed out before thread terminated.

`EINVAL` `abstime` value is invalid (`tv_sec` is less than 0 or `tv_nsec` is greater than 1e9).

`pthread_timedjoin_np()` never returns the error `EINTR`.

### VERSIONS

These functions first appeared in glibc in version 2.3.3.

### ATTRIBUTES

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

??

?Interface ? Attribute ? Value ?

??

?`pthread_tryjoin_np()`, ? Thread safety ? MT-Safe ?

?`pthread_timedjoin_np()` ? ? ?

??

## CONFORMING TO

These functions are nonstandard GNU extensions; hence the suffix "\_np" (nonportable) in the names.

## EXAMPLES

The following code waits to join for up to 5 seconds:

```

struct timespec ts;

int s;

...

if (clock_gettime(CLOCK_REALTIME, &ts) == -1) {
    /* Handle error */
}

ts.tv_sec += 5;

s = pthread_timedjoin_np(thread, NULL, &ts);

if (s != 0) {
    /* Handle error */
}

```

## BUGS

The pthread\_timedjoin\_np() function measures time by internally calculating a relative sleep interval that is then measured against the CLOCK\_MONOTONIC clock instead of the CLOCK\_REALTIME clock. Consequently, the timeout is unaffected by discontinuous changes to the CLOCK\_REALTIME clock.

## SEE ALSO

clock\_gettime(2), pthread\_exit(3), pthread\_join(3), pthreads(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/>.