## NAME

pthread_yield − yield the processor

## SYNOPSIS

```
#define _GNU_SOURCE         /* See feature_test_macros(7) */
#include <pthread.h>

int pthread_yield(void);
```

Compile and link with *−pthread*.

## DESCRIPTION

**pthread_yield**() causes the calling thread to relinquish the CPU. The thread is placed at the end of the run queue for its static priority and another thread is scheduled to run. For further details, see **sched_yield**(2)

## RETURN VALUE

On success, **pthread_yield**() returns 0; on error, it returns an error number.

## ERRORS

On Linux, this call always succeeds (but portable and future-proof applications should nevertheless handle a possible error return).

## ATTRIBUTES

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

| Interface | Attribute | Value |
|---|---|---|
| **pthread_yield**() | Thread safety | MT-Safe |

## CONFORMING TO

This call is nonstandard, but present on several other systems. Use the standardized **sched_yield**(2) instead.

## NOTES

On Linux, this function is implemented as a call to **sched_yield**(2).

**pthread_yield**() is intended for use with real-time scheduling policies (i.e., **SCHED_FIFO** or **SCHED_RR**). Use of **pthread_yield**() with nondeterministic scheduling policies such as **SCHED_OTHER** is unspecified and very likely means your application design is broken.

## SEE ALSO

**sched_yield**(2), **pthreads**(7), **sched**(7)

## COLOPHON

This page is part of release 5.05 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/.