

**NAME**

pthread\_yield – yield the processor

**LIBRARY**

POSIX threads library (*libpthread*, *-lpthread*)

**SYNOPSIS**

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

[[deprecated]] int pthread_yield(void);
```

**DESCRIPTION**

**Note:** This function is deprecated; see below.

**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).

**VERSIONS**

Since glibc 2.34, this function is marked as deprecated.

**ATTRIBUTES**

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

Interface	Attribute	Value
<b>pthread_yield()</b>	Thread safety	MT-Safe

**STANDARDS**

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)**