

**NAME**

`pthread_rwlockattr_setkind_np`, `pthread_rwlockattr_getkind_np` – set/get the read-write lock kind of the thread read-write lock attribute object

**SYNOPSIS**

```
#include <pthread.h>
```

```
int pthread_rwlockattr_setkind_np(pthread_rwlockattr_t *attr,
                                   int pref);
int pthread_rwlockattr_getkind_np(const pthread_rwlockattr_t *attr,
                                   int *pref);
```

Compile and link with `-pthread`.

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

```
pthread_rwlockattr_setkind_np(), pthread_rwlockattr_getkind_np():
    _XOPEN_SOURCE >= 500 || _POSIX_C_SOURCE >= 200809L
```

**DESCRIPTION**

The `pthread_rwlockattr_setkind_np()` function sets the "lock kind" attribute of the read-write lock attribute object referred to by *attr* to the value specified in *pref*. The argument *pref* may be set to one of the following:

**PTHREAD\_RWLOCK\_PREFER\_READER\_NP**

This is the default. A thread may hold multiple read locks; that is, read locks are recursive. According to The Single Unix Specification, the behavior is unspecified when a reader tries to place a lock, and there is no write lock but writers are waiting. Giving preference to the reader, as is set by **PTHREAD\_RWLOCK\_PREFER\_READER\_NP**, implies that the reader will receive the requested lock, even if a writer is waiting. As long as there are readers, the writer will be starved.

**PTHREAD\_RWLOCK\_PREFER\_WRITER\_NP**

This is intended as the write lock analog of **PTHREAD\_RWLOCK\_PREFER\_READER\_NP**. This is ignored by glibc because the POSIX requirement to support recursive writer locks would cause this option to create trivial deadlocks; instead use **PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP** which ensures the application developer will not take recursive read locks thus avoiding deadlocks.

**PTHREAD\_RWLOCK\_PREFER\_WRITER\_NONRECURSIVE\_NP**

Setting the lock kind to this avoids writer starvation as long as any read locking is not done in a recursive fashion.

The `pthread_rwlockattr_getkind_np()` function returns the value of the lock kind attribute of the read-write lock attribute object referred to by *attr* in the pointer *pref*.

**RETURN VALUE**

On success, these functions return 0. Given valid pointer arguments, `pthread_rwlockattr_getkind_np()` always succeeds. On error, `pthread_rwlockattr_setkind_np()` returns a nonzero error number.

**ERRORS****EINVAL**

*pref* specifies an unsupported value.

**VERSIONS**

The `pthread_rwlockattr_getkind_np()` and `pthread_rwlockattr_setkind_np()` functions first appeared in glibc 2.1.

**CONFORMING TO**

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

**SEE ALSO**

`pthreads(7)`

**COLOPHON**

This page is part of release 5.02 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/>.