## NAME

pthread.h - threads

## SYNOPSIS

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

## DESCRIPTION

The <*pthread.h*> header defines the following symbols:

```
PTHREAD_CANCEL_ASYNCHRONOUS
PTHREAD_CANCEL_ENABLE
PTHREAD_CANCEL_DEFERRED
PTHREAD_CANCEL_DISABLE
PTHREAD_CANCELED
PTHREAD_COND_INITIALIZER
PTHREAD_CREATE_DETACHED
PTHREAD_CREATE_JOINABLE
PTHREAD_EXPLICIT_SCHED
PTHREAD_INHERIT_SCHED
PTHREAD_MUTEX_DEFAULT
PTHREAD_MUTEX_ERRORCHECK
PTHREAD_MUTEX_NORMAL
PTHREAD_MUTEX_INITIALIZER
PTHREAD_MUTEX_RECURSIVE
PTHREAD_ONCE_INIT
PTHREAD_PRIO_INHERIT
PTHREAD_PRIO_NONE
PTHREAD_PRIO_PROTECT
PTHREAD_PROCESS_SHARED
PTHREAD_PROCESS_PRIVATE
PTHREAD_RWLOCK_INITIALIZER
PTHREAD_SCOPE_PROCESS
PTHREAD_SCOPE_SYSTEM
```

The **pthread_attr_t**, **pthread_cond_t**, **pthread_condattr_t**, **pthread_key_t**, **pthread_mutex_t**, **pthread_mutexattr_t**, **pthread_once_t**, **pthread_rwlock_t**, **pthread_rwlockattr_t** and **pthread_t** types are defined as described in *<sys/types.h>*.

The following are declared as functions and may also be declared as macros. Function prototypes must be provided for use with an ISO C compiler.

```
int   pthread_attr_destroy(pthread_attr_t *);
int   pthread_attr_getdetachstate(const pthread_attr_t *, int *);
int   pthread_attr_getguardsize(const pthread_attr_t *, size_t *);
int   pthread_attr_getinheritsched(const pthread_attr_t *, int *);
int   pthread_attr_getschedparam(const pthread_attr_t *,
```

```
        struct sched_param *);
int   pthread_attr_getschedpolicy(const pthread_attr_t *, int *);
int   pthread_attr_getscope(const pthread_attr_t *, int *);
int   pthread_attr_getstackaddr(const pthread_attr_t *, void **);
int   pthread_attr_getstacksize(const pthread_attr_t *, size_t *);
int   pthread_attr_init(pthread_attr_t *);
int   pthread_attr_setdetachstate(pthread_attr_t *, int);
int   pthread_attr_setguardsize(pthread_attr_t *, size_t);
int   pthread_attr_setinheritsched(pthread_attr_t *, int);
int   pthread_attr_setschedparam(pthread_attr_t *,
        const struct sched_param *);
int   pthread_attr_setschedpolicy(pthread_attr_t *, int);
int   pthread_attr_setscope(pthread_attr_t *, int);
int   pthread_attr_setstackaddr(pthread_attr_t *, void *);
int   pthread_attr_setstacksize(pthread_attr_t *, size_t);
int   pthread_cancel(pthread_t);
void  pthread_cleanup_push(void*), void *);
void  pthread_cleanup_pop(int);
int   pthread_cond_broadcast(pthread_cond_t *);
int   pthread_cond_destroy(pthread_cond_t *);
int   pthread_cond_init(pthread_cond_t *, const pthread_condattr_t *);
int   pthread_cond_signal(pthread_cond_t *);
int   pthread_cond_timedwait(pthread_cond_t *,
        pthread_mutex_t *, const struct timespec *);
int   pthread_cond_wait(pthread_cond_t *, pthread_mutex_t *);
int   pthread_condattr_destroy(pthread_condattr_t *);
int   pthread_condattr_getpshared(const pthread_condattr_t *, int *);
int   pthread_condattr_init(pthread_condattr_t *);
int   pthread_condattr_setpshared(pthread_condattr_t *, int);
int   pthread_create(pthread_t *, const pthread_attr_t *,
        void *(*)(void *), void *);
int   pthread_detach(pthread_t);
int   pthread_equal(pthread_t, pthread_t);
void  pthread_exit(void *);
int   pthread_getconcurrency(void);
int   pthread_getschedparam(pthread_t, int *, struct sched_param *);
void *pthread_getspecific(pthread_key_t);
int   pthread_join(pthread_t, void **);
int   pthread_key_create(pthread_key_t *, void (*)(void *));
int   pthread_key_delete(pthread_key_t);
int   pthread_mutex_destroy(pthread_mutex_t *);
int   pthread_mutex_getprioceiling(const pthread_mutex_t *, int *);
int   pthread_mutex_init(pthread_mutex_t *, const pthread_mutexattr_t *);
int   pthread_mutex_lock(pthread_mutex_t *);
int   pthread_mutex_setprioceiling(pthread_mutex_t *, int, int *);
int   pthread_mutex_trylock(pthread_mutex_t *);
int   pthread_mutex_unlock(pthread_mutex_t *);
int   pthread_mutexattr_destroy(pthread_mutexattr_t *);
int   pthread_mutexattr_getprioceiling(const pthread_mutexattr_t *,
        int *);
int   pthread_mutexattr_getprotocol(const pthread_mutexattr_t *, int *);
int   pthread_mutexattr_getpshared(const pthread_mutexattr_t *, int *);
int   pthread_mutexattr_gettype(const pthread_mutexattr_t *, int *);
int   pthread_mutexattr_init(pthread_mutexattr_t *);
int   pthread_mutexattr_setprioceiling(pthread_mutexattr_t *, int);
int   pthread_mutexattr_setprotocol(pthread_mutexattr_t *, int);
int   pthread_mutexattr_setpshared(pthread_mutexattr_t *, int);
int   pthread_mutexattr_settype(pthread_mutexattr_t *, int);
int   pthread_once(pthread_once_t *, void (*)(void));
int   pthread_rwlock_destroy(pthread_rwlock_t *);
int   pthread_rwlock_init(pthread_rwlock_t *,
        const pthread_rwlockattr_t *);
int   pthread_rwlock_rdlock(pthread_rwlock_t *);
int   pthread_rwlock_tryrdlock(pthread_rwlock_t *);
int   pthread_rwlock_trywrlock(pthread_rwlock_t *);
int   pthread_rwlock_unlock(pthread_rwlock_t *);
int   pthread_rwlock_wrlock(pthread_rwlock_t *);
int   pthread_rwlockattr_destroy(pthread_rwlockattr_t *);
int   pthread_rwlockattr_getpshared(const pthread_rwlockattr_t *,
        int *);
```

```
int   pthread_rwlockattr_init(pthread_rwlockattr_t *);
int   pthread_rwlockattr_setpshared(pthread_rwlockattr_t *, int);
pthread_t
      pthread_self(void);
int   pthread_setcancelstate(int, int *);
int   pthread_setcanceltype(int, int *);
int   pthread_setconcurrency(int);
int   pthread_setschedparam(pthread_t, int ,
          const struct sched_param *);
int   pthread_setspecific(pthread_key_t, const void *);
void  pthread_testcancel(void);
```

Inclusion of the <*pthread.h*> header will make visible symbols defined in the headers *<sched.h>* and *<time.h>*.

## APPLICATION USAGE

An interpretation request has been filed with IEEE PASC concerning requirements for visibility of symbols in this header.

## FUTURE DIRECTIONS

None.

## SEE ALSO

*pthread_attr_init()*, *pthread_attr_getguardsize()*, *pthread_attr_setscope()*, *pthread_cancel()*, *pthread_cleanup_push()*, *pthread_cond_init()*, *pthread_cond_signal()*, *pthread_cond_wait()*, *pthread_condattr_init()*, *pthread_create()*, *pthread_detach()*, *pthread_equal()*, *pthread_exit()*, *pthread_getconcurrency()*, *pthread_getschedparam()*, *pthread_join()*, *pthread_key_create()*, *pthread_key_delete()*, *pthread_mutex_init()*, *pthread_mutex_lock()*, *pthread_mutex_setprioceiling()*, *pthread_mutexattr_init()*, *pthread_mutexattr_gettype()*, *pthread_mutexattr_setprotocol()*, *pthread_once()*, *pthread_self()*, *pthread_setcancelstate()*, *pthread_setspecific()*, *pthread_rwlock_init()*, *pthread_rwlock_rdlock()*, *pthread_rwlock_unlock()*, *pthread_rwlock_wrlock()*, *pthread_rwlockattr_init()*, *<sched.h>*, *<time.h>*.