

NAME

pthread_attr_init, **pthread_attr_destroy**, **pthread_attr_setstack**, **pthread_attr_getstack**,
pthread_attr_setstacksize, **pthread_attr_getstacksize**, **pthread_attr_setguardsize**,
pthread_attr_getguardsize, **pthread_attr_setstackaddr**, **pthread_attr_getstackaddr**,
pthread_attr_setdetachstate, **pthread_attr_getdetachstate**, **pthread_attr_setinheritsched**,
pthread_attr_getinheritsched, **pthread_attr_setschedparam**, **pthread_attr_getschedparam**,
pthread_attr_setschedpolicy, **pthread_attr_getschedpolicy**, **pthread_attr_setscope**, **pthread_attr_getscope**
- thread attribute operations

LIBRARY

POSIX Threads Library (libpthread, -lpthread)

SYNOPSIS

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

int

```
pthread_attr_init(pthread_attr_t *attr);
```

int

```
pthread_attr_destroy(pthread_attr_t *attr);
```

int

```
pthread_attr_setstack(pthread_attr_t *attr, void *stackaddr, size_t stacksize);
```

int

```
pthread_attr_getstack(const pthread_attr_t * restrict attr, void ** restrict stackaddr,  
size_t * restrict stacksize);
```

int

```
pthread_attr_setstacksize(pthread_attr_t *attr, size_t stacksize);
```

int

```
pthread_attr_getstacksize(const pthread_attr_t * restrict attr, size_t * restrict stacksize);
```

int

```
pthread_attr_setguardsize(pthread_attr_t *attr, size_t guardsize);
```

int

```
pthread_attr_getguardsize(const pthread_attr_t * restrict attr, size_t * restrict guardsize);
```

*int***pthread_attr_setstackaddr**(*pthread_attr_t *attr, void *stackaddr*);*int***pthread_attr_getstackaddr**(*const pthread_attr_t *attr, void **stackaddr*);*int***pthread_attr_setdetachstate**(*pthread_attr_t *attr, int detachstate*);*int***pthread_attr_getdetachstate**(*const pthread_attr_t *attr, int *detachstate*);*int***pthread_attr_setinheritsched**(*pthread_attr_t *attr, int inheritsched*);*int***pthread_attr_getinheritsched**(*const pthread_attr_t *restrict attr, int *restrict inheritsched*);*int***pthread_attr_setschedparam**(*pthread_attr_t *attr, const struct sched_param *param*);*int***pthread_attr_getschedparam**(*const pthread_attr_t *attr, struct sched_param *param*);*int***pthread_attr_setschedpolicy**(*pthread_attr_t *attr, int policy*);*int***pthread_attr_getschedpolicy**(*const pthread_attr_t *restrict attr, int *restrict policy*);*int***pthread_attr_setscope**(*pthread_attr_t *attr, int contentionscope*);*int***pthread_attr_getscope**(*const pthread_attr_t *restrict attr, int *restrict contentionscope*);

DESCRIPTION

Thread attributes are used to specify parameters to **pthread_create**(*pthread_create*(*pthread_create*)). One attribute object can be used in multiple calls to **pthread_create**(*pthread_create*), with or without modifications between calls.

The **pthread_attr_init()** function initializes *attr* with all the default thread attributes.

The **pthread_attr_destroy()** function destroys *attr*.

The **pthread_attr_set*()** functions set the attribute that corresponds to each function name.

The **pthread_attr_get*()** functions copy the value of the attribute that corresponds to each function name to the location pointed to by the second function parameter.

RETURN VALUES

If successful, these functions return 0. Otherwise, an error number is returned to indicate the error.

ERRORS

The **pthread_attr_init()** function will fail if:

[ENOMEM] Out of memory.

The **pthread_attr_destroy()** function will fail if:

[EINVAL] Invalid value for *attr*.

The **pthread_attr_setstacksize()** and **pthread_attr_setstack()** functions will fail if:

[EINVAL] *stacksize* is less than PTHREAD_STACK_MIN.

The **pthread_attr_setdetachstate()** function will fail if:

[EINVAL] Invalid value for *detachstate*.

The **pthread_attr_setinheritsched()** function will fail if:

[EINVAL] Invalid value for *attr*.

The **pthread_attr_setschedparam()** function will fail if:

[EINVAL] Invalid value for *attr*.

[ENOTSUP] Invalid value for *param*.

The **pthread_attr_setschedpolicy()** function will fail if:

[EINVAL] Invalid value for *attr*.

[ENOTSUP] Invalid or unsupported value for *policy*.

The **pthread_attr_setscope()** function will fail if:

[EINVAL] Invalid value for *attr*.

[ENOTSUP] Invalid or unsupported value for *contentionscope*.

SEE ALSO

`pthread_attr_affinity_np(3)`, `pthread_attr_get_np(3)`, `pthread_create(3)`

STANDARDS

pthread_attr_init(), **pthread_attr_destroy()**, **pthread_attr_setstacksize()**, **pthread_attr_getstacksize()**, **pthread_attr_setstackaddr()**, **pthread_attr_getstackaddr()**, **pthread_attr_setdetachstate()**, and **pthread_attr_getdetachstate()** functions conform to ISO/IEC 9945-1:1996 ("POSIX.1")

The **pthread_attr_setinheritsched()**, **pthread_attr_getinheritsched()**, **pthread_attr_setschedparam()**, **pthread_attr_getschedparam()**, **pthread_attr_setschedpolicy()**, **pthread_attr_getschedpolicy()**, **pthread_attr_setscope()**, and **pthread_attr_getscope()** functions conform to Version 2 of the Single UNIX Specification ("SUSv2")