NAME

pthread_attr_setstacksize, pthread_attr_getstacksize − set/get stack size attribute in thread attributes object

SYNOPSIS

#include <pthread.h>

int pthread_attr_setstacksize(pthread_attr_t ****attr*, size_t** *stacksize***);**
int pthread_attr_getstacksize(const pthread_attr_t ****attr*, size_t ****stacksize***);**

Compile and link with *−pthread*.

DESCRIPTION

The **pthread_attr_setstacksize**() function sets the stack size attribute of the thread attributes object referred to by *attr* to the value specified in *stacksize*.

The stack size attribute determines the minimum size (in bytes) that will be allocated for threads created using the thread attributes object *attr*.

The **pthread_attr_getstacksize**() function returns the stack size attribute of the thread attributes object referred to by *attr* in the buffer pointed to by *stacksize*.

RETURN VALUE

On success, these functions return 0; on error, they return a nonzero error number.

ERRORS

pthread_attr_setstacksize() can fail with the following error:

EINVAL
The stack size is less than **PTHREAD_STACK_MIN** (16384) bytes.

On some systems, **pthread_attr_setstacksize**() can fail with the error **EINVAL** if *stacksize* is not a multiple of the system page size.

VERSIONS

These functions are provided by glibc since version 2.1.

ATTRIBUTES

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

Interface	Attribute	Value
pthread_attr_setstacksize(), **pthread_attr_getstacksize**()	Thread safety	MT-Safe

CONFORMING TO

POSIX.1-2001, POSIX.1-2008.

NOTES

For details on the default stack size of new threads, see **pthread_create**(3).

A thread's stack size is fixed at the time of thread creation. Only the main thread can dynamically grow its stack.

The **pthread_attr_setstack**(3) function allows an application to set both the size and location of a caller-allocated stack that is to be used by a thread.

BUGS

As at glibc 2.8, if the specified *stacksize* is not a multiple of **STACK_ALIGN** (16 bytes on most architectures), it may be rounded *downward*, in violation of POSIX.1, which says that the allocated stack will be at least *stacksize* bytes.

EXAMPLE

See **pthread_create**(3).

SEE ALSO

getrlimit(2), **pthread_attr_init**(3), **pthread_attr_setguardsize**(3), **pthread_attr_setstack**(3), **pthread_create**(3), **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/.