

**NAME**

pthread\_attr\_setstackaddr, pthread\_attr\_getstackaddr - set/get stack address attribute in thread attributes object

**SYNOPSIS**

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

```
int pthread_attr_setstackaddr(pthread_attr_t *attr, void *stackaddr);
int pthread_attr_getstackaddr(const pthread_attr_t *attr, void **stackaddr);
```

Compile and link with *-pthread*.

**DESCRIPTION**

These functions are obsolete: **do not use them**. Use [pthread\\_attr\\_setstack\(3\)](#) and [pthread\\_attr\\_getstack\(3\)](#) instead.

The [pthread\\_attr\\_setstackaddr\(\)](#) function sets the stack address attribute of the thread attributes object referred to by *attr* to the value specified in *stackaddr*. This attribute specifies the location of the stack that should be used by a thread that is created using the thread attributes object *attr*.

*stackaddr* should point to a buffer of at least `PTHREAD_STACK_MIN` bytes that was allocated by the caller. The pages of the allocated buffer should be both readable and writable.

The [pthread\\_attr\\_getstackaddr\(\)](#) function returns the stack address attribute of the thread attributes object referred to by *attr* in the buffer pointed to by *stackaddr*.

**RETURN VALUE**

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

**ERRORS**

No errors are defined (but applications should nevertheless handle a possible error return).

**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
<a href="#">pthread_attr_setstackaddr()</a> , <a href="#">pthread_attr_getstackaddr()</a>	Thread safety	MT-Safe

**CONFORMING TO**

POSIX.1-2001 specifies these functions but marks them as obsolete. POSIX.1-2008 removes the specification of these functions.

**NOTES**

*Do not use these functions!* They cannot be portably used, since they provide no way of specifying the direction of growth or the range of the stack. For example, on architectures with a stack that grows downward, *stackaddr* specifies the next address past the *highest* address of the allocated stack area. However, on architectures with a stack that grows upward, *stackaddr* specifies the *lowest* address in the allocated stack area. By contrast, the *stackaddr* used by [pthread\\_attr\\_setstack\(3\)](#) and [pthread\\_attr\\_getstack\(3\)](#), is always a pointer to the lowest address in the allocated stack area (and the *stacksize* argument specifies the range of the stack).

**SEE ALSO**

[pthread\\_attr\\_init\(3\)](#), [pthread\\_attr\\_setstack\(3\)](#), [pthread\\_attr\\_setstacksize\(3\)](#), [pthread\\_create\(3\)](#), [pthreads\(7\)](#)

**COLOPHON**

This page is part of release 4.10 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/>.