

**NAME**

pthread\_detach – detach a thread

**LIBRARY**

POSIX threads library (*libpthread*, *-lpthread*)

**SYNOPSIS**

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

```
int pthread_detach(pthread_t thread);
```

**DESCRIPTION**

The **pthread\_detach()** function marks the thread identified by *thread* as detached. When a detached thread terminates, its resources are automatically released back to the system without the need for another thread to join with the terminated thread.

Attempting to detach an already detached thread results in unspecified behavior.

**RETURN VALUE**

On success, **pthread\_detach()** returns 0; on error, it returns an error number.

**ERRORS****EINVAL**

*thread* is not a joinable thread.

**ESRCH**

No thread with the ID *thread* could be found.

**ATTRIBUTES**

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

Interface	Attribute	Value
<b>pthread_detach()</b>	Thread safety	MT-Safe

**STANDARDS**

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

**NOTES**

Once a thread has been detached, it can't be joined with **pthread\_join(3)** or be made joinable again.

A new thread can be created in a detached state using **pthread\_attr\_setdetachstate(3)** to set the detached attribute of the *attr* argument of **pthread\_create(3)**.

The detached attribute merely determines the behavior of the system when the thread terminates; it does not prevent the thread from being terminated if the process terminates using **exit(3)** (or equivalently, if the main thread returns).

Either **pthread\_join(3)** or **pthread\_detach()** should be called for each thread that an application creates, so that system resources for the thread can be released. (But note that the resources of any threads for which one of these actions has not been done will be freed when the process terminates.)

**EXAMPLES**

The following statement detaches the calling thread:

```
pthread_detach(pthread_self());
```

**SEE ALSO**

**pthread\_attr\_setdetachstate(3)**, **pthread\_cancel(3)**, **pthread\_create(3)**, **pthread\_exit(3)**, **pthread\_join(3)**,  **pthreads(7)**