

**NAME**

sem\_destroy – destroy an unnamed semaphore

**LIBRARY**

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

**SYNOPSIS**

```
#include <semaphore.h>

int sem_destroy(sem_t *sem);
```

**DESCRIPTION**

**sem\_destroy()** destroys the unnamed semaphore at the address pointed to by *sem*.

Only a semaphore that has been initialized by **sem\_init(3)** should be destroyed using **sem\_destroy()**.

Destroying a semaphore that other processes or threads are currently blocked on (in **sem\_wait(3)**) produces undefined behavior.

Using a semaphore that has been destroyed produces undefined results, until the semaphore has been reinitialized using **sem\_init(3)**.

**RETURN VALUE**

**sem\_destroy()** returns 0 on success; on error, -1 is returned, and *errno* is set to indicate the error.

**ERRORS****EINVAL**

*sem* is not a valid semaphore.

**ATTRIBUTES**

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

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

**STANDARDS**

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

**NOTES**

An unnamed semaphore should be destroyed with **sem\_destroy()** before the memory in which it is located is deallocated. Failure to do this can result in resource leaks on some implementations.

**SEE ALSO**

**sem\_init(3)**, **sem\_post(3)**, **sem\_wait(3)**, **sem\_overview(7)**