

NAME

`sem_post` - unlock a semaphore

SYNOPSIS

```
#include <semaphore.h>

int sem_post(sem_t *sem);
```

Link with `-pthread`.

DESCRIPTION

`sem_post()` increments (unlocks) the semaphore pointed to by `sem`. If the semaphore's value consequently becomes greater than zero, then another process or thread blocked in a `sem_wait(3)` call will be woken up and proceed to lock the semaphore.

RETURN VALUE

`sem_post()` returns 0 on success; on error, the value of the semaphore is left unchanged, -1 is returned, and `errno` is set to indicate the error.

ERRORS**EINVAL**

`sem` is not a valid semaphore.

E_OVERFLOW

The maximum allowable value for a semaphore would be exceeded.

ATTRIBUTES

For an explanation of the terms used in this section, see [attributes\(7\)](#).

Interface	Attribute	Value
<code>sem_post()</code>	Thread safety	MT-Safe

CONFORMING TO

POSIX.1-2001.

NOTES

`sem_post()` is async-signal-safe: it may be safely called within a signal handler.

EXAMPLE

See [sem_wait\(3\)](#).

SEE ALSO

[sem_getvalue\(3\)](#), [sem_wait\(3\)](#), [sem_overview\(7\)](#), [signal-safety\(7\)](#)

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/>.