



## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sem\_post.3p' command***

***\$ man sem\_post.3p***

SEM\_POST(3P)      POSIX Programmer's Manual      SEM\_POST(3P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

sem\_post ? unlock a semaphore

### SYNOPSIS

```
#include <semaphore.h>

int sem_post(sem_t *sem);
```

### DESCRIPTION

The sem\_post() function shall unlock the semaphore referenced by sem by performing a semaphore unlock operation on that semaphore.

If the semaphore value resulting from this operation is positive, then no threads were blocked waiting for the semaphore to become unlocked; the semaphore value is simply incremented.

If the value of the semaphore resulting from this operation is zero, then one of the threads blocked waiting for the semaphore shall be allowed to return successfully from its call to sem\_wait(). If the Process Scheduling option is supported, the thread to be unblocked shall be chosen in a manner appropriate to the scheduling policies and parameters in effect for the blocked threads. In the case of the sched?

ulers SCHED\_FIFO and SCHED\_RR, the highest priority waiting thread shall be unblocked, and if there is more than one highest priority thread blocked waiting for the semaphore, then the highest priority thread that has been waiting the longest shall be unblocked. If the Process Scheduling option is not defined, the choice of a thread to unblock is unspecified.

If the Process Sporadic Server option is supported, and the scheduling policy is SCHED\_SPORADIC, the semantics are as per SCHED\_FIFO above.

The sem\_post() function shall be async-signal-safe and may be invoked from a signal-catching function.

## RETURN VALUE

If successful, the sem\_post() function shall return zero; otherwise, the function shall return -1 and set errno to indicate the error.

## ERRORS

The sem\_post() function may fail if:

EINVAL The sem argument does not refer to a valid semaphore.

The following sections are informative.

## EXAMPLES

See sem\_timedwait().

## APPLICATION USAGE

None.

## RATIONALE

None.

## FUTURE DIRECTIONS

None.

## SEE ALSO

semctl(), semget(), semop(), sem\_timedwait(), sem\_trywait()

The Base Definitions volume of POSIX.1?2017, Section 4.12, Memory Synchronization, <semaphore.h>

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specific?

cations Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

IEEE/The Open Group

2017

SEM\_POST(3P)