



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sem_init.3p' command

\$ man sem_init.3p

SEM_INIT(3P) POSIX Programmer's Manual SEM_INIT(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_init ? initialize an unnamed semaphore

SYNOPSIS

```
#include <semaphore.h>

int sem_init(sem_t *sem, int pshared, unsigned value);
```

DESCRIPTION

The `sem_init()` function shall initialize the unnamed semaphore referred to by `sem`. The value of the initialized semaphore shall be `value`. Following a successful call to `sem_init()`, the semaphore may be used in subsequent calls to `sem_wait()`, `sem_timedwait()`, `sem_trywait()`, `sem_post()`, and `sem_destroy()`. This semaphore shall remain usable until the semaphore is destroyed.

If the `pshared` argument has a non-zero value, then the semaphore is shared between processes; in this case, any process that can access the semaphore `sem` can use `sem` for performing `sem_wait()`, `sem_timedwait()`, `sem_trywait()`, `sem_post()`, and `sem_destroy()` operations.

If the `pshared` argument is zero, then the semaphore is shared between

threads of the process; any thread in this process can use sem for per?
forming sem_wait(), sem_timedwait(), sem_trywait(), sem_post(), and
sem_destroy() operations.

See Section 2.9.9, Synchronization Object Copies and Alternative Map?
pings for further requirements.

Attempting to initialize an already initialized semaphore results in
undefined behavior.

RETURN VALUE

Upon successful completion, the sem_init() function shall initialize
the semaphore in sem and return 0. Otherwise, it shall return -1 and
set errno to indicate the error.

ERRORS

The sem_init() function shall fail if:

EINVAL The value argument exceeds {SEM_VALUE_MAX}.

ENOSPC A resource required to initialize the semaphore has been ex?
hausted, or the limit on semaphores ({SEM_NSEMS_MAX}) has been
reached.

EPERM The process lacks appropriate privileges to initialize the sema?
phore.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

sem_destroy(), sem_post(), sem_timedwait(), sem_trywait()

The Base Definitions volume of POSIX.1?2017, <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 Specifications 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 .

IEEE/The Open Group

2017

SEM_INIT(3P)