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 'shm_unlink.3p' command

$ man shm_unlink.3p

SHM_UNLINK(3P) POSIX Programmer's Manual SHM_UNLINK(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

shm_unlink ? remove a shared memory object (REALTIME)

SYNOPSIS

```
#include <sys/mman.h>
int shm_unlink(const char *name);
```

DESCRIPTION

The shm_unlink() function shall remove the name of the shared memory object named by the string pointed to by name.

If one or more references to the shared memory object exist when the object is unlinked, the name shall be removed before shm_unlink() re? turns, but the removal of the memory object contents shall be postponed until all open and map references to the shared memory object have been removed.

Even if the object continues to exist after the last shm_unlink(), re? use of the name shall subsequently cause shm_open() to behave as if no shared memory object of this name exists (that is, shm_open() will fail if O_CREAT is not set, or will create a new shared memory object if

O_CREAT is set).

RETURN VALUE

Upon successful completion, a value of zero shall be returned. Other?
wise, a value of -1 shall be returned and errno set to indicate the er?
ror. If -1 is returned, the named shared memory object shall not be
changed by this function call.

ERRORS

The shm_unlink() function shall fail if:

EACCES Permission is denied to unlink the named shared memory object.

ENOENT The named shared memory object does not exist.

The shm_unlink() function may fail if:

ENAMETOOLONG

The length of the name argument exceeds {_POSIX_PATH_MAX} on systems that do not support the XSI option or exceeds {_XOPEN_PATH_MAX} on XSI systems, or has a pathname component that is longer than {_POSIX_NAME_MAX} on systems that do not support the XSI option or longer than {_XOPEN_NAME_MAX} on XSI systems. A call to shm_unlink() with a name argument that con? tains the same shared memory object name as was previously used in a successful shm_open() call shall not give an [ENAMETOOLONG] error.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Names of memory objects that were allocated with open() are deleted with unlink() in the usual fashion. Names of memory objects that were allocated with shm_open() are deleted with shm_unlink(). Note that the actual memory object is not destroyed until the last close and unmap on it have occurred if it was already in use.

RATIONALE

None.

FUTURE DIRECTIONS

A future version might require the shm_open() and shm_unlink() func?
tions to have semantics similar to normal file system operations.

SEE ALSO

close(), mmap(), munmap(), shmat(), shmctl(), shmdt(), shm_open()

The Base Definitions volume of POSIX.1?2017, <sys_mman.h>

COPYRIGHT