NAME

msync - synchronize a mapped region

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/mman.h>

int
msync(*void *addr, size_t len, int flags*);

DESCRIPTION

The **msync**() system call writes any modified pages back to the file system and updates the file modification time. If *len* is 0, all modified pages within the region containing *addr* will be flushed; if *len* is non-zero, only those pages containing *addr* and *len-1* succeeding locations will be examined. The *flags* argument may be specified as follows:

MS_ASYNC	Return immediately
MS_SYNC	Perform synchronous writes
MS_INVALIDATE	Invalidate all cached data

RETURN VALUES

The **msync**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **msync**() system call will fail if:

[EBUSY]	Some or all of the pages in the specified region are locked and MS_INVALIDATE is specified.
[EINVAL]	The *addr* argument is not a multiple of the hardware page size.
[ENOMEM]	The addresses in the range starting at *addr* and continuing for *len* bytes are outside the range allowed for the address space of a process or specify one or more pages that are not mapped.
[EINVAL]	The *flags* argument was both MS_ASYNC and MS_INVALIDATE. Only one of these flags is allowed.

[EIO] An error occurred while writing at least one of the pages in the specified region.

SEE ALSO

madvise(2), mincore(2), mlock(2), mprotect(2), munmap(2)

HISTORY

The **msync**() system call first appeared in 4.4BSD.

BUGS

The **msync**() system call is usually not needed since BSD implements a coherent file system buffer cache. However, it may be used to associate dirty VM pages with file system buffers and thus cause them to be flushed to physical media sooner rather than later.