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

msync — synchronize memory with physical storage

SYNOPSIS

#include <sys/mman.h>

int msync(void *_addr_, size_t _len_, int _flags_);

DESCRIPTION

The *msync*() function shall write all modified data to permanent storage locations, if any, in those whole pages containing any part of the address space of the process starting at address *addr* and continuing for *len* bytes. If no such storage exists, *msync*() need not have any effect. If requested, the *msync*() function shall then invalidate cached copies of data.

The implementation may require that *addr* be a multiple of the page size as returned by *sysconf*().

For mappings to files, the *msync*() function shall ensure that all write operations are completed as defined for synchronized I/O data integrity completion. It is unspecified whether the implementation also writes out other file attributes. When the *msync*() function is called on MAP_PRIVATE mappings, any modified data shall not be written to the underlying object and shall not cause such data to be made visible to other processes. It is unspecified whether data in MAP_PRIVATE mappings has any permanent storage locations. The effect of *msync*() on a shared memory object or a typed memory object is unspecified. The behavior of this function is unspecified if the mapping was not established by a call to *mmap*().

The *flags* argument is constructed from the bitwise-inclusive OR of one or more of the following flags defined in the <*sys/mman.h*> header:

Symbolic Constant	Description
MS_ASYNC	Perform asynchronous writes.
MS_SYNC	Perform synchronous writes.
MS_INVALIDATE	Invalidate cached data.

When MS_ASYNC is specified, *msync*() shall return immediately once all the write operations are initiated or queued for servicing; when MS_SYNC is specified, *msync*() shall not return until all write operations are completed as defined for synchronized I/O data integrity completion. Either MS_ASYNC or MS_SYNC shall be specified, but not both.

When MS_INVALIDATE is specified, *msync*() shall invalidate all cached copies of mapped data that are inconsistent with the permanent storage locations such that subsequent references shall obtain data that was consistent with the permanent storage locations sometime between the call to *msync*() and the first subsequent memory reference to the data.

If *msync*() causes any write to a file, the file's last data modification and last file status change timestamps shall be marked for update.

RETURN VALUE

Upon successful completion, *msync*() shall return 0; otherwise, it shall return −1 and set *errno* to indicate the error.

ERRORS

The *msync*() function shall fail if:

EBUSY

Some or all of the addresses in the range starting at *addr* and continuing for *len* bytes are locked, and MS_INVALIDATE is specified.

EINVAL

The value of *flags* is invalid.

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.

The *msync*() function may fail if:

EINVAL

The value of *addr* is not a multiple of the page size as returned by *sysconf*().

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The *msync*() function is only supported if the Synchronized Input and Output option is supported, and thus need not be available on all implementations.

The *msync*() function should be used by programs that require a memory object to be in a known state; for example, in building transaction facilities.

Normal system activity can cause pages to be written to disk. Therefore, there are no guarantees that *msync*() is the only control over when pages are or are not written to disk.

RATIONALE

The *msync*() function writes out data in a mapped region to the permanent storage for the underlying object. The call to *msync*() ensures data integrity of the file.

After the data is written out, any cached data may be invalidated if the MS_INVALIDATE flag was specified. This is useful on systems that do not support read/write consistency.

FUTURE DIRECTIONS

None.

SEE ALSO

mmap( ), *sysconf*( )

The Base Definitions volume of POSIX.1-2008, **<sys_mman.h>**

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2013 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, Copyright (C) 2013 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. (This is POSIX.1-2008 with the 2013 Technical Corrigendum 1 applied.) 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.unix.org/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 .