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 'fsync.3p' command

$ man fsync.3p

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

fsync ? synchronize changes to a file

SYNOPSIS

#include <unistd.h>

int fsync(int fildes);

DESCRIPTION

The fsync() function shall request that all data for the open file de? scriptor named by fildes is to be transferred to the storage device as? sociated with the file described by fildes. The nature of the transfer is implementation-defined. The fsync() function shall not return until the system has completed that action or until an error is detected.

If _POSIX_SYNCHRONIZED_IO is defined, the fsync() function shall force all currently queued I/O operations associated with the file indicated by file descriptor fildes to the synchronized I/O completion state. All I/O operations shall be completed as defined for synchronized I/O file integrity completion.

RETURN VALUE

Upon successful completion, fsync() shall return 0. Otherwise, -1 shall be returned and errno set to indicate the error. If the fsync() function fails, outstanding I/O operations are not guaranteed to have been completed.

ERRORS

The fsync() function shall fail if:

EBADF The fildes argument is not a valid descriptor.

EINTR The fsync() function was interrupted by a signal.

EINVAL The fildes argument does not refer to a file on which this operation is possible.

EIO An I/O error occurred while reading from or writing to the file system.

In the event that any of the queued I/O operations fail, fsync() shall return the error conditions defined for read() and write().

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The fsync() function should be used by programs which require modifications to a file to be completed before continuing; for example, a program which contains a simple transaction facility might use it to ensure that all modifications to a file or files caused by a transaction are recorded.

RATIONALE

The fsync() function is intended to force a physical write of data from the buffer cache, and to assure that after a system crash or other failure that all data up to the time of the fsync() call is recorded on the disk. Since the concepts of ``buffer cache'', ``system crash'', ``physical write'', and ``non-volatile storage'' are not defined here, the wording has to be more abstract.

If _POSIX_SYNCHRONIZED_IO is not defined, the wording relies heavily on the conformance document to tell the user what can be expected from the system. It is explicitly intended that a null implementation is permit?

ted. This could be valid in the case where the system cannot assure non-volatile storage under any circumstances or when the system is highly fault-tolerant and the functionality is not required. In the middle ground between these extremes, fsync() might or might not actu? ally cause data to be written where it is safe from a power failure.

The conformance document should identify at least that one configura? tion exists (and how to obtain that configuration) where this can be assured for at least some files that the user can select to use for critical data. It is not intended that an exhaustive list is required, but rather sufficient information is provided so that if critical data needs to be saved, the user can determine how the system is to be con? figured to allow the data to be written to non-volatile storage.

It is reasonable to assert that the key aspects of fsync() are unrea? sonable to test in a test suite. That does not make the function any less valuable, just more difficult to test. A formal conformance test should probably force a system crash (power shutdown) during the test for this condition, but it needs to be done in such a way that auto? mated testing does not require this to be done except when a formal record of the results is being made. It would also not be unreasonable to omit testing for fsync(), allowing it to be treated as a quality-of-implementation issue.

FUTURE DIRECTIONS

None.

SEE ALSO

sync()

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

COPYRIGHT

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.ker?nel.org/doc/man-pages/reporting_bugs.html .