fsync - Man Page

synchronize changes to a file

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.

Synopsis

```
#include <unistd.h>

int fsync(int fildes);
```

Description

The *fsync*() function shall request that all data for the open file descriptor named by *fildes* is to be transferred to the storage device associated 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.

fsync - Man Page

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.

fsync - Man Page

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 permitted. 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 actually cause data to be written where it is safe from a power failure. The conformance document should identify at least that one configuration 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 configured to allow the data to be written to non-volatile storage.

fsync - Man Page

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
automated 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

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

fsync - Man Page

Referenced By

aio_fsync(3p), aio.h(0p), fdatasync(3p), sync(3p), unistd.h(0p).

2017 IEEE/The Open Group POSIX Programmer's Manual

Home   Blog   About