

# fsync - Man Page

*синхронизация изменений в файле*

## Пролог

Эта страница руководства является частью Руководства программиста POSIX. Реализация этого интерфейса в Linux может отличаться (обратитесь к соответствующей странице руководства Linux для получения подробной информации о поведении Linux), или интерфейс может быть не реализован в Linux.

## Краткий обзор

#включить <unistd.h>

```
int fsync (int fildes);
```

## Описание

Функция *fsync()* запрашивает передачу всех данных для открытого файлового дескриптора, названного *fildes*, на запоминающее устройство, связанное с файлом, описанным *fildes*. Характер передачи определяется реализацией. Функция *fsync()* не возвращается до тех пор, пока система не завершит это действие или пока не будет обнаружена ошибка.

Если `_POSIX_SYNCHRONIZED_IO` определен, функция *fsync()* должна принудительно перевести все операции ввода-вывода, находящиеся в очереди, связанные с файлом, указанным *fildes* *файлового дескриптора*, в состояние завершения синхронизированного ввода-вывода. Все операции ввода-вывода должны быть завершены, как определено для завершения целостности синхронизированного файла ввода-вывода.

# fsync - Man Page

После успешного завершения *fsync()* возвращает 0. В противном случае возвращается значение -1 и устанавливается значение *errno* для указания ошибки. Если функция *fsync()* выходит из строя, выполнение незавершенных операций ввода-вывода не гарантируется.

## Ошибки

Функция *fsync()* завершится ошибкой, если:

### **EBADF**

Аргумент *fildev* не является допустимым дескриптором.

### **EINTR**

Функция *fsync()* была прервана сигналом.

### **EINVAL**

Аргумент *fildev* не ссылается на файл, с которым возможна эта операция.

**ЭИО** При чтении или записи в файловую систему произошла ошибка ввода-вывода.

В случае сбоя любой из операций ввода-вывода в очереди функция *fsync()* возвращает условия ошибки, определенные для *read()* и *write()*.

*Следующие разделы являются информативными.*

## Примеры

Нет.

# fsync - Man Page

Функция *fsync()* должна использоваться программами, которые требуют, чтобы изменения файла были завершены перед продолжением; например, программа, содержащая простое средство транзакции, может использовать его для обеспечения записи всех изменений файла или файлов, вызванных транзакцией.

## Обоснование

Функция *fsync()* предназначена для принудительной физической записи данных из буферного кэша и обеспечения того, чтобы после сбоя системы или другого сбоя все данные до момента *вызова* *fsync()* были записаны на диск. Поскольку понятия “буферный кэш”, “сбой системы”, “физическая запись” и “энергонезависимое хранилище” здесь не определены, формулировка должна быть более абстрактной.

Если `_POSIX_SYNCHRONIZED_IO` не определен, формулировка в значительной степени зависит от документа соответствия, чтобы сообщить пользователю, что можно ожидать от системы. Явно подразумевается, что реализация `null` разрешена. Это может быть справедливо в случае, когда система не может обеспечить энергонезависимое хранилище ни при каких обстоятельствах или когда система очень отказоустойчива и функциональность не требуется. В середине между этими крайностями *fsync()* может или не может фактически привести к записи данных там, где они защищены от сбоя питания. В документе соответствия должно быть указано, что существует хотя бы одна конфигурация (и как получить эту конфигурацию), где это может быть гарантировано по крайней мере для некоторых файлов, которые пользователь может выбрать для использования для критических данных. Предполагается, что не требуется исчерпывающий список, а достаточно информации, чтобы, если необходимо сохранить критические данные, пользователь мог определить, как система должна быть настроена для записи данных в энергонезависимое хранилище.

# fsync - Man Page

ее сложнее протестировать. Формальным тест на соответствие, вероятно, должен вызвать сбой системы (отключение питания) во время теста для этого условия, но это должно быть сделано таким образом, чтобы автоматическое тестирование не требовало этого, за исключением случаев, когда делается формальная запись результатов. Также было бы неразумно опустить тестирование для *fsync()*, что позволяет рассматривать его как проблему качества реализации.

## Будущие направления

Нет.

## См . также

[синхронизация\(\)](#)

Базовый том определений POSIX.1-2017, [<unistd.h>](#)

## Авторские права

Части этого текста перепечатаны и воспроизведены в электронном виде из 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 GroupГруппа. В случае любого несоответствия между этой версией и исходным стандартом IEEE и Open Group исходный стандарт IEEE и Open Group является документом рефери. Оригинальный стандарт можно получить онлайн по адресу <http://www.opengroup.org/unix/online.html> .

Любые типографские ошибки или ошибки форматирования, которые появляются на этой странице, скорее всего, были введены во время преобразования

# fsync - Man Page

## Ссылка на

[aio\\_fsync \(3p\)](#), [aio.h \(0p\)](#), [fdatasync \(3p\)](#), [синхронизация \(3p\)](#),  
[unistd.h\(0p\)](#).

2017 IEEE/The Open Group POSIX Programmer's Manual