## NAME

sync, syncfs – commit filesystem caches to disk

## SYNOPSIS

**#include <unistd.h>**

**void sync(void);**

**int syncfs(int** *fd***);**

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

**sync**():
_XOPEN_SOURCE >= 500 || /* Since glibc 2.19: */ _DEFAULT_SOURCE || /* Glibc versions <= 2.19: */ _BSD_SOURCE

**syncfs**():
_GNU_SOURCE

## DESCRIPTION

**sync**() causes all pending modifications to filesystem metadata and cached file data to be written to the underlying filesystems.

**syncfs**() is like **sync**(), but synchronizes just the filesystem containing file referred to by the open file descriptor *fd*.

## RETURN VALUE

**syncfs**() returns 0 on success; on error, it returns −1 and sets *errno* to indicate the error.

## ERRORS

**sync**() is always successful.

**syncfs**() can fail for at least the following reason:

**EBADF**
*fd* is not a valid file descriptor.

## VERSIONS

**syncfs**() first appeared in Linux 2.6.39; library support was added to glibc in version 2.14.

## CONFORMING TO

**sync**(): POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD.

**syncfs**() is Linux-specific.

## NOTES

Since glibc 2.2.2, the Linux prototype for **sync**() is as listed above, following the various standards. In glibc 2.2.1 and earlier, it was "int sync(void)", and **sync**() always returned 0.

According to the standard specification (e.g., POSIX.1-2001), **sync**() schedules the writes, but may return before the actual writing is done. However Linux waits for I/O completions, and thus **sync**() or **syncfs**() provide the same guarantees as fsync called on every file in the system or filesystem respectively.

## BUGS

Before version 1.3.20 Linux did not wait for I/O to complete before returning.

## SEE ALSO

sync(1), fdatasync(2), fsync(2)

## COLOPHON

This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man–pages/.