NAME

sync, syncfs − commit filesystem caches to disk

LIBRARY

Standard C library (*libc*, *−lc*)

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 <= 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 reasons:

EBADF
fd is not a valid file descriptor.

EIO An error occurred during synchronization. This error may relate to data written to any file on the filesystem, or on metadata related to the filesystem itself.

ENOSPC
Disk space was exhausted while synchronizing.

ENOSPC, **EDQUOT**
Data was written to a file on NFS or another filesystem which does not allocate space at the time of a **write**(2) system call, and some previous write failed due to insufficient storage space.

VERSIONS

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

STANDARDS

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.

In mainline kernel versions prior to Linux 5.8, **syncfs**() will fail only when passed a bad file descriptor

(**EBADF**). Since Linux 5.8,**syncfs**() will also report an error if one or more inodes f ailed to be written back since the last **syncfs**() call.

BUGS

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

SEE ALSO

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