NAME

sendfile − transfer data between file descriptors

SYNOPSIS

#include <sys/sendfile.h>

ssize_t sendfile(int *out_fd***, int** *in_fd***, off_t ****offset***, size_t** *count***);**

DESCRIPTION

sendfile() copies data between one file descriptor and another. Because this copying is done within the kernel, **sendfile**() is more efficient than the combination of **read**(2) and **write**(2), which would require transferring data to and from user space.

in_fd should be a file descriptor opened for reading and *out_fd* should be a descriptor opened for writing.

If *offset* is not NULL, then it points to a variable holding the file offset from which **sendfile**() will start reading data from *in_fd*. When **sendfile**() returns, this variable will be set to the offset of the byte following the last byte that was read. If *offset* is not NULL, then **sendfile**() does not modify the current file offset of *in_fd*; otherwise the current file offset is adjusted to reflect the number of bytes read from *in_fd*.

count is the number of bytes to copy between the file descriptors.

Presently (Linux 2.6.9): *in_fd*, must correspond to a file which supports **mmap**(2)-like operations (i.e., it cannot be a socket); and *out_fd* must refer to a socket.

Applications may wish to fall back to **read**(2)/**write**(2) in the case where **sendfile**() fails with **EINVAL** or **ENOSYS**.

RETURN VALUE

If the transfer was successful, the number of bytes written to *out_fd* is returned. On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EAGAIN
Non-blocking I/O has been selected using **O_NONBLOCK** and the write would block.

EBADF
The input file was not opened for reading or the output file was not opened for writing.

EFAULT
Bad address.

EINVAL
Descriptor is not valid or locked, or an **mmap**(2)-like operation is not available for *in_fd*.

EIO Unspecified error while reading from *in_fd*.

ENOMEM
Insufficient memory to read from *in_fd*.

VERSIONS

sendfile() is a new feature in Linux 2.2. The include file <*sys/sendfile.h*> is present since glibc 2.1.

CONFORMING TO

Not specified in POSIX.1-2001, or other standards.

Other Unix systems implement **sendfile**() with different semantics and prototypes. It should not be used in portable programs.

NOTES

If you plan to use **sendfile**() for sending files to a TCP socket, but need to send some header data in front of the file contents, you will find it useful to employ the **TCP_CORK** option, described in **tcp**(7), to minimize the number of packets and to tune performance.

In Linux 2.4 and earlier, *out_fd* could refer to a regular file, and **sendfile**() changed the current offset of that file.

SEE ALSO

mmap(2), **open**(2), **socket**(2), **splice**(2)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.