## NAME

fifo – first-in first-out special file, named pipe

## DESCRIPTION

A FIFO special file (a named pipe) is similar to a pipe, except that it is accessed as part of the file system. It can be opened by multiple processes for reading or writing. When processes are exchanging data via the FIFO, the kernel passes all data internally without writing it to the file system. Thus, the FIFO special file has no contents on the file system; the file system entry merely serves as a reference point so that processes can access the pipe using a name in the file system.

The kernel maintains exactly one pipe object for each FIFO special file that is opened by at least one process. The FIFO must be opened on both ends (reading and writing) before data can be passed. Normally, opening the FIFO blocks until the other end is opened also.

A process can open a FIFO in non-blocking mode. In this case, opening for read only will succeed even if no-one has opened on the write side yet, opening for write only will fail with **ENXIO** (no such device or address) unless the other end has already been opened.

Under Linux, opening a FIFO for read and write will succeed both in blocking and non-blocking mode. POSIX leaves this behavior undefined. This can be used to open a FIFO for writing while there are no readers available. A process that uses both ends of the connection in order to communicate with itself should be very careful to avoid deadlocks.

## NOTES

When a process tries to write to a FIFO that is not opened for read on the other side, the process is sent a **SIGPIPE** signal.

FIFO special files can be created by **mkfifo**(3), and are indicated by *ls −l* with the file type 'p'.

## SEE ALSO

**mkfifo**(1), **open**(2), **pipe**(2), **sigaction**(2), **signal**(2), **socketpair**(2), **mkfifo**(3), **pipe**(7)

## 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/.