

# mkfifo(3) — Linux manual page

[NAME](#) | [SYNOPSIS](#) | [DESCRIPTION](#) | [RETURN VALUE](#) | [ERRORS](#) | [VERSIONS](#) | [ATTRIBUTES](#) | [CONFORMING TO](#) | [SEE ALSO](#) | [COLOPHON](#)

**MKFIFO(3)****Linux Programmer's Manual****MKFIFO(3)**

## NAME [top](#)

mkfifo, mkfifoat – создать специальный файл FIFO (именованный канал)

## СИНОПСИС [сверху](#)

```
#include <sys/types.h>
#include <sys/stat.h>
```

```
int mkfifo(const char *pathname, mode_t mode);
```

```
#include <fcntl.h> /* Определение констант AT_* */
#include <sys/stat.h>
```

```
int mkfifoat(int dirfd, const char *pathname, mode_t mode);
```

Требования к макросу Feature Test для glibc (см. [feature\\_test\\_macros\(7\)](#)):

```
mkfifoat():
```

Начиная с glibc 2.10:  
\_POSIX\_C\_SOURCE >= 200809L  
До glibc 2.10:  
\_ATFILE\_SOURCE

## ОПИСАНИЕ [top](#)

**mkfifo()** создает специальный файл FIFO с именем *pathname*. *режим* указывает разрешения FIFO. Он модифицируется **umask** **процесса** обычным способом: разрешения созданного файла (*mode* & ~*umask*).

Специальный файл FIFO похож на канал, за исключением того, что он создается по-другому. Вместо того, чтобы быть анонимным каналом связи, специальный файл FIFO вводится в файловую систему путем вызова **mkfifo()**.

После того, как вы создали специальный файл FIFO таким образом, любой процесс может открыть его для чтения или записи, так же, как обычный файл. Однако он должен быть открыт на обоих концах одновременно, прежде чем вы сможете приступить к выполнению каких-либо операций ввода или вывода на нем. Открытие FIFO для чтения обычно блокируется до тех пор, пока какой-либо другой процесс не откроет тот же FIFO для записи, и наоборот. См. [fifo\(7\)](#) для неблокирующей обработки специальных файлов FIFO.

### **mkfifoat()**

Функция **mkfifoat()** работает точно так же, как **mkfifo()**, за исключением различий, описанных здесь.

Если путь, указанный в *pathname*, является относительным, то он интерпретируется относительно каталога, на который ссылается файловый дескриптор *dirfd* (а не относительно текущего рабочего каталога вызывающего процесса, как это делается **mkfifo()** для относительного пути).

Если *pathname* является относительным, а *dirfd* – специальным значением **AT\_FDCWD**, то *pathname* интерпретируется относительно текущего рабочего каталога вызывающего процесса (например, **mkfifo()** ).

Если путь абсолютный, то *dirfd* игнорируется.

См. [openat\(2\)](#) для объяснения необходимости **mkfifoat()** .

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ [top](#)

При успешном **выполнении** **mkfifo()** и **mkfifoat()** возвращают 0. При ошибке возвращается значение -1, а значение *errno* указывает на ошибку.

## ОШИБКИ [сверху](#)

**EACCES** Один из каталогов в *pathname* не разрешил поиск разрешение (выполнить).

Путь **EBADF** (**mkfifoat()**) относителен, *но dirfd* не является ни **AT\_FDCWD**, ни допустимым файловым дескриптором.

**EDQUOT** Пользовательская квота дисковых блоков или инодов в файловой системе исчерпана.

*Путь EEXIST* уже существует. Это включает в себя случай, когда *pathname* – это символическая ссылка, висящая или нет.

### **ENAMETOOLONG**

Либо общая длина *pathname* больше

**PATH\_MAX**, или отдельный компонент имени файла имеет длину больше **NAME\_MAX**. В системе GNU нет ограничений на общую длину имени файла, но некоторые файловые системы могут устанавливать ограничения на длину компонента.

**ENOENT** Компонент каталога в *pathname* не существует или является висячей символической ссылкой.

**ENOSPC** В каталоге или файловой системе нет места для нового файла.

### **ENOTDIR**

Компонент, используемый в качестве каталога в *pathname*, на самом деле не является каталогом.

### **ENOTDIR**

(**mkfifoat**()) *pathname* – это относительный путь, а *dirfd* – это дескриптор файла, ссылающийся на файл, отличный от каталога.

*Путь EROFS* относится к файловой системе, доступной только для чтения.

## **ВЕРСИИ** [top](#)

**mkfifoat**() был добавлен в glibc в версии 2.4. Он реализован с помощью **mknodat**(2), доступного в Linux начиная с ядра 2.6.16.

## **АТРИБУТЫ** [top](#)

Объяснение терминов, используемых в этом разделе, см. в разделе [Атрибуты \(7\)](#).

Интерфейс	Атрибут	Значение		
<b>mkfifo</b> (), <b>mkfifoat</b> ()	Безопасность резьбы	MT-Safe		

**СООТВЕТСТВУЕТ** [top](#)

**mkfifo()**: POSIX.1-2001, POSIX.1-2008.

**mkfifoat()**: POSIX.1-2008.

**СМ. ТАКЖЕ** [top](#)

[mkfifo \(1\)](#), [закрыть\(2\)](#), [открыто \(2\)](#), [читать\(2\)](#), [stat\(2\)](#), [umask\(2\)](#),  
[запись\(2\)](#), [fifo\(7\)](#)

**КОЛОФОН** [top](#)

Эта страница является частью версии 5.13 проекта Linux *man-pages*.

Описание проекта, информацию о сообщениях об ошибках  
и последнюю версию этой страницы можно найти по  
адресу: <https://www.kernel.org/doc/man-pages/>.

**GNU 2021-08-27 MKFIFO (3)**

---

Страницы, которые ссылаются на эту страницу: [mkfifo\(1\)](#), [mknod\(2\)](#), [open\(2\)](#), [umask\(2\)](#),  
[unlink\(2\)](#), [remove\(3\)](#), [fifo\(7\)](#), [pipe\(7\)](#), [signal-safety\(7\)](#)

---

[Авторские права и лицензия на эту страницу руководства](#)

---

HTML-рендеринг создан 2021-08-27 [Майклом Керриском](#), автором *интерфейса  
программирования Linux*, сопровождающим проекта Linux *man-pages*.

Подробнее о углубленных **курсах системного программирования Linux / UNIX**,  
которые я преподаю, читайте [здесь](#).

Хостинг от [jambit GmbH](#).

