

chmod(3p) — Linux manual page

[PROLOG](#) | [NAME](#) | [SYNOPSIS](#) | [DESCRIPTION](#) | [RETURN VALUE](#) | [ERRORS](#) | [EXAMPLES](#) |
[APPLICATION USAGE](#) | [RATIONALE](#) | [FUTURE DIRECTIONS](#) | [SEE ALSO](#) | [COPYRIGHT](#)

CHMOD(3P)**POSIX Programmer's Manual****CHMOD(3P)****PROLOG**[top](#)

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME[top](#)

chmod, fchmodat – change mode of a file

SYNOPSIS[top](#)

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

```
int chmod(const char *path, mode_t mode);
```

```
#include <fcntl.h>
```

```
int fchmodat(int fd, const char *path, mode_t mode, int flag);
```

DESCRIPTION [top](#)

Функция *chmod*() должна изменить S_ISUID, S_ISGID, S_ISVTX и биты разрешения файла файла, названного именем пути, на которое указывает аргумент *path*, на соответствующие биты в аргументе *mode*. Приложение должно убедиться, что эффективный идентификатор пользователя процесса соответствует владельцу файла или что процесс имеет соответствующие привилегии для этого.

S_ISUID, S_ISGID, S_ISVTX и биты разрешения файла описаны в *<sys/stat.h>*.

Если вызывающий процесс не имеет соответствующих привилегий, и если идентификатор группы файла не совпадает с эффективным идентификатором группы или одним из дополнительных идентификаторов группы, и если файл является обычным файлом, бит S_ISGID (set-group-ID при выполнении) в режиме файла должен быть очищен после успешного возврата из *chmod*().

Дополнительные ограничения, определяемые реализацией, могут привести к игнорированию битов S_ISUID и S_ISGID в *режиме*.

После успешного завершения *chmod*() помечает для обновления последнюю метку времени изменения статуса файла.

Функция *fchmodat*() должна быть эквивалентна функции *chmod*(), за исключением случая, когда *path* указывает относительный путь.

В этом случае файл подлежащий изменению определяется относительно каталога связанного с файловым дескриптором *fd* вместо текущего рабочего каталога. Если режим доступа к описанию открытого файла, связанному с файловым дескриптором, не является O_SEARCH, функция должна проверить, разрешен ли поиск в каталоге с использованием текущих разрешений каталога, лежащего в

основе файлового дескриптора. Если режим доступа – `O_SEARCH`, функция не будет выполнять проверку.

Значения для *флага* строятся побитовым включением ИЛИ из флагов из следующего списка, определенного в *<fcntl.h>*:

`AT_SYMLINK_NOFOLLOW`

Если *путь* называет символическую ссылку, затем изменяется режим символической ссылки.

Если *fchmodat()* передается специальное значение `AT_FDCWD` в *fd* параметр, должен использоваться текущий рабочий каталог. Если также *флаг* равен нулю, поведение должно быть идентичным вызову *chmod()*.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ [top](#)

После успешного завершения эти функции должны вернуть 0.

В противном случае эти функции должны возвращать -1 и устанавливать *errno* для указания ошибки. Если возвращается значение -1, изменение режима файла не происходит.

ОШИБКИ [сверху](#)

Эти функции не будут работать, если:

Разрешение на поиск `EACCES` отказано компоненту префикса пути

.

ELOOP Цикл существует в символических ссылках, встречающихся при разрешении аргумента *path*.

ENAMETOOLONG

Длина компонента пути больше

{NAME_MAX}.

ENOENT Компонент *path* не называет существующий файл или *путь* это пустая строка.

ENOTDIR

Компонент префикса *path* называет существующий файл, который не является ни каталогом, ни символической ссылкой на каталог, или аргумент *path* содержит по крайней мере один не-<косая черта> символ и заканчивается одним или несколькими завершающими <slash> символы и последний путь компонент именуется существующий файл, который не является ни каталогом, ни символической ссылкой на каталог.

EPERM Действующий идентификатор пользователя не совпадает с владельцем файла, и процесс не имеет соответствующих привилегий.

EROFS Именованный файл находится в файловой системе, доступной только для чтения.

Функция *fchmodat()* завершится ошибкой, если:

Режим доступа к описанию открытого файла, связанного с *fd*, не является `O_SEARCH`, а разрешения каталога, лежащего *в основе fd*, не разрешают поиск в каталоге.

EBADF Аргумент *path* не указывает абсолютный путь, а *fd* аргумент не является ни `AT_FDCWD`, ни допустимым файловым дескриптором, открытым для чтения или поиска.

ENOTDIR

Аргумент *path* не является абсолютным путем, а *fd* является файловым дескриптором, связанным с файлом, не относящимся к каталогу.

Эти функции могут выйти из строя, если:

EINTR Сигнал был пойман во время выполнения функции.

EINVAL Значение аргумента *mode* недопустимо.

При разрешении аргумента *path* было обнаружено более символических ссылок {SYMLoop_MAX}

.

ENAMETOOLONG

Длина пути превышает {PATH_MAX}, или разрешение пути символической ссылки дало промежуточный результат с длиной, превышающей {PATH_MAX} .

Функция *fchmodat()* может выйти из строя, если:

EINVAL Значение аргумента *flag* недопустимо.

EOPNOTSUPP

Бит AT_SYMLINK_NOFOLLOW устанавливается в аргументе *flag*, *path* называет символическую ссылку, и система не поддерживает изменение режима символической ссылки.

Следующие разделы являются информативными.

ПРИМЕРЫ [сверху](#)

Настройка разрешений на чтение для пользователя, группы и других

Следующий пример устанавливает разрешения на чтение для владельца, группы и других.

```
#включить <sys/stat.h>
```

```
const char *path;
```

```
...
```

```
chmod(путь, S_IRUSR|S_IRGRP|S_IROTH);
```

Настройка разрешений на чтение, запись и выполнение только для владельца

В следующем примере устанавливаются разрешения на чтение, запись и выполнение для владельца и разрешения "нет" для группы и других.

```
#включить <sys/stat.h>
```

```
const char *path;
```

```
...
```

```
chmod(path, S_IRWXU);
```

Настройка различных разрешений для владельца, группы и других

В следующем примере устанавливаются разрешения владельца для чтения, записи и выполнения ФАЙЛА ИЗМЕНЕНИЙ, разрешения группы для чтения и выполнения и другие разрешения для чтения.

```
#включить <sys/stat.h>
```

```
#определить ФАЙЛ ИЗМЕНЕНИЙ "/etc/myfile"
```

```
...
```

```
chmod(CHANGFILE, S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH);
```

Настройка и проверка прав доступа к файлам

В следующем примере устанавливаются биты разрешения файла для файла с именем **/home/cnd/mod1**, а затем вызывается *stat()* функция для проверки разрешений.

```
#включить <sys/types.h>
```

```
#включить <sys/stat.h>
```

```
int status;
```

```
struct stat buffer
```

```
...
```

```
chmod("/home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
```

```
status = stat("/home/cnd/mod1", &buffer);
```

ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЙ [top](#)

Чтобы убедиться, что биты `S_ISUID` и `S_ISGID` установлены, приложение, требующее этого, должно использовать `stat()` после успешного `chmod()` для проверки этого.

Любые файловые дескрипторы, открытые в данный момент любым процессом в файле, могут стать недействительными, если режим файла будет изменен на значение, которое запретит доступ к этому процессу. Одна из ситуаций, когда это может произойти, – это файловая система без состояния. Такое поведение не будет происходить в соответствующей среде.

ОБОСНОВАНИЕ [сверху](#)

Этот том POSIX.1-2017 указывает, что бит `S_ISGID` очищается `chmod()` в обычном файле при определенных условиях.

Это указано исходя из предположения, что обычные файлы могут быть выполнены, и система должна запретить пользователям создавать исполняемый `setgid()` файлы работают с привилегиями, которых нет у вызывающего абонента. В реализациях, поддерживающих выполнение других типов файлов, бит `S_ISGID` должен быть очищен для этих типов файлов при тех же обстоятельствах.

Реализации, использующие бит `S_ISUID` для указания какой-либо другой функции (например, обязательной блокировки записей) в неисполняемых файлах, не должны очищать этот бит при записи. Они должны очистить бит для исполняемых файлов и любых других случаев, когда бит предоставляет специальные полномочия процессам, которые изменяют файл. Содержание. Аналогичные комментарии применяются к бит `S_ISGID`.

Цель функции `fchmodat()` состоит в том, чтобы включить изменение режима файлов в каталогах, отличных от текущего рабочего каталога, без воздействия условий гонки. Любая часть пути к файлу может быть изменена параллельно вызову `chmod()`,

что приведет к неопределенному поведению. Открыв файловый дескриптор для целевого каталога и используя *fchmodat* функция (*()*) можно гарантировать, что измененный файл находится относительно нужного каталога. Некоторые реализации могут позволить изменить режим символических ссылок. Это не поддерживается интерфейсами в спецификации POSIX. Системы с такой поддержкой предоставляют интерфейс с именем *lchmod()*. Для поддержки таких реализаций *fchmodat()* имеет параметр флага.

FUTURE DIRECTIONS [top](#)

Нет.

СМ. ТАКЖЕ [top](#)

[access\(3p\)](#), [chown\(3p\)](#), [exec\(1p\)](#), [fstatat\(3p\)](#), [fstatvfs \(3p\)](#),
[mkdir \(3p\)](#), [mkfifo \(3p\)](#), [mknod \(3p\)](#), [open \(3p\)](#)

Объем базовых определений POSIX.1–2017, [fcntl.h\(0p\)](#),
[sys_stat.h\(0p\)](#), [sys_types.h\(0p\)](#)

КОПИРАЙТ [top](#)

Части этого текста перепечатаны и воспроизведены в электронном виде из стандарта IEEE Std 1003.1–2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc и Open Group. В случае любого расхождения между этой версией и исходным стандартом IEEE и Open Group исходный стандарт IEEE и Open Group

Стандарт – это документ рефери. Оригинальный стандарт можно получить онлайн по адресу <http://www.opengroup.org/unix/online.html> .

Любые типографские ошибки или ошибки форматирования, которые появляются на этой странице, скорее всего, были допущены во время преобразования исходных файлов в формат man-страницы. Чтобы сообщить о таких ошибках, см. https://www.kernel.org/doc/man-pages/reporting_bugs.html.

IEEE/The Open Group 2017 CHMOD (3P)

Страницы, которые ссылаются на эту страницу: [sys_stat.h\(0p\)](#), [chmod \(1p\)](#), [доступ \(3p\)](#), [chown\(3p\)](#), [exec\(3p\)](#), [fchmod\(3p\)](#), [fchmodat\(3p\)](#), [fstatat\(3p\)](#), [fstatvfs\(3p\)](#), [lockf\(3p\)](#), [mkdir\(3p\)](#), [mkfifo \(3p\)](#), [mknod\(3p\)](#), [открыть \(3p\)](#), [posix_spawn\(3p\)](#), [запись \(3p\)](#)

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

Для получения подробной информации о углубленных **курсах обучения системному программированию Linux / UNIX**, которые я преподаю, посмотрите [здесь](#).

Хостинг от [jambit GmbH](#).

