

lockf - Man Page

блокировка записей в файлах

Пролог

Эта страница руководства является частью Руководства программиста POSIX. Реализация этого интерфейса в Linux может отличаться (обратитесь к соответствующей странице руководства Linux для получения подробной информации о поведении Linux), или интерфейс может быть не реализован в Linux.

Краткое описание

#включить <unistd.h>

```
int lockf(int fildes, int function, off_t size);
```

Описание

Функция *lockf()* блокирует разделы файла блокировками консультативного режима. Вызовы *lockf()* из потоков других процессов, которые пытаются заблокировать заблокированный раздел файла, должны либо возвращать значение ошибки, либо блокировать его до тех пор, пока раздел не станет разблокированным. Все блокировки для процесса удаляются, когда процесс завершается. Блокировка записей с помощью функции *lockf()* поддерживается для обычных файлов и может поддерживаться для других файлов.

Аргумент *fildes* – это открытый файловый дескриптор. Чтобы установить блокировку с помощью этой функции, файловый дескриптор должен быть открыт с разрешением только на запись (O_WRONLY) или с разрешением на чтение / запись (O_RDWR).

lockf - Man Page

функции определяются в `<unistd.h>` следующим образом:

Функция	Описание
F_ULOCK	Разблокируйте заблокированные секции.
F_LOCK	Заприте секцию для исключительного использования.
F_TLOCK	Проверьте и заблокируйте раздел для эксклюзивного использования.
F_TEST	Проверьте раздел на наличие блокировок другими процессами.

F_TEST должен определить, присутствует ли блокировка другим процессом в указанном разделе.

F_LOCK и F_TLOCK должны блокировать раздел файла, если этот раздел доступен.

F_ULOCK удаляет блокировки из раздела файла.

Аргумент *size* – это количество смежных байтов, которые должны быть заблокированы или разблокированы. Блокируемый или разблокируемый раздел начинается с текущего смещения в файле и продолжается вперед для положительного размера или назад для отрицательного размера (предыдущие байты до, но не включая текущее смещение). Если *размер* равен 0, то участок от текущего смещения до максимально возможного смещения файла должен быть заблокирован (то есть от текущего смещения до настоящего или любого будущего конца файла). Область не должна быть выделена для файла, который будет заблокирован, потому что блокировки могут существовать после конца файла.

Секции, заблокированные с помощью F_LOCK или F_TLOCK, могут полностью или частично содержать или содержаться ранее заблокированной секцией для того же процесса. Когда это происходит или если возникают соседние заблокированные секции, секции должны быть объединены в одну

lockf - Man Page

неудачно.

Запросы F_LOCK и F_TLOCK отличаются только действием, выполняемым, если раздел недоступен. F_LOCK блокирует вызывающий поток до тех пор, пока раздел не станет доступным. F_TLOCK приведет к сбою функции, если раздел уже заблокирован другим процессом.

Блокировки файлов должны быть освобождены при первом закрытии процессом блокировки любого файлового дескриптора для файла.

Запросы F_ULOCK могут освободить (полностью или частично) один или несколько заблокированных разделов, контролируемых процессом. Заблокированные разделы должны быть разблокированы, начиная с текущего смещения файла через байты размера или до конца файла, если *размер* равен (`off_t`)0. Когда вся заблокированная секция не освобождается (то есть, когда начало или конец области, подлежащей разблокировке, попадает в заблокированную секцию), остальные части этой секции остаются заблокированными процессом. Освобождение центральной части заблокированной секции приведет к тому, что оставшиеся заблокированные начальная и конечная части станут двумя отдельными заблокированными секциями. Если запрос приведет к тому, что количество блокировок в системе превысит установленный системой предел, запрос завершится неудачно.

Вероятность взаимоблокировки возникает, если потоки процесса, управляющего заблокированной секцией, блокируются путем доступа к заблокированной секции другого процесса. Если система обнаружит, что произойдет взаимоблокировка, *функция lockf()* завершится **ошибкой [EDEADLK]**.

Взаимодействие между *блокировками fcntl()* и *lockf()* не определено.

Блокировка на секции должна быть прервана любым сигналом.

lockf - Man Page

объекта типа `FILE`, когда процесс имеет существующую блокировку, *размер* которой равен 0 и которая включает последний байт запрашиваемого раздела, должен рассматриваться как запрос `F_ULOCK`. запрос на разблокировку из начала запрашиваемого раздела с размером, равным 0. В противном случае запрос `F_ULOCK` попытается разблокировать только запрошенный раздел.

Попытка заблокировать раздел файла, связанный с буферизованным потоком, приводит к неопределенным результатам.

Возвращаемое значение

После успешного завершения `lockf()` возвращает 0. В противном случае он вернет -1, установит `errno` для указания на ошибку, и существующие блокировки не будут изменены.

Ошибки

Функция `lockf()` завершится неудачей, если:

EBADF

Аргумент *fildev* не является допустимым открытым файловым дескриптором; или *функция* `F_LOCK` или `F_TLOCK` и *fildev* не является допустимым файловым дескриптором, открытым для записи.

EACCES или EAGAIN

Аргумент *функции* – `F_TLOCK` или `F_TEST`, и раздел уже заблокирован другим процессом.

EDEADLK

Аргумент *функции* – `F_LOCK`, и обнаруживается взаимоблокировка.

EINTR

lockf - Man Page

Аргумент *функции* не является одним из F_LOCK, F_TLOCK, F_TEST или F_ULOCK; или *размер* плюс текущее смещение файла меньше 0.

EOVERFLOW

Смещение первого или, если *размер* не равен 0, то последнего байта в запрашиваемом разделе не может быть корректно представлено в объекте типа **off_t**.

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

EAGAIN

Аргумент *функции* – F_LOCK или F_TLOCK, и файл сопоставляется с помощью *ttar()* .

EDEADLK или ENOLCK

Аргументом *функции* является F_LOCK, F_TLOCK или F_ULOCK, и запрос приведет к тому, что количество блокировок превысит установленный системой предел.

EOPNOTSUPP или EINVAL

Реализация не поддерживает блокировку файлов типа, указанного аргументом *files*.

Следующие разделы носят информативный характер.

Примеры

Блокировка части файла

lockf - Man Page

процесса. Заблокированы только первые 10000 байт, и вызов блокировки завершается неудачно, если другой процесс уже заблокировал какую-либо часть этой области.

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

int fildes;
statuc int;

...
fildes = open("/home/cnd/mod1", O_RDWR);
status = lockf(fildes, F_TLOCK, (off_t)10000);
```

Применение

Блокировка записи не должна использоваться в сочетании с функциями *fopen()*, *fread()*, *fwrite()* и другими функциями *stdio*. Вместо этого следует использовать более примитивные, не буферизованные функции (такие как *open()*). Неожиданные результаты могут возникнуть в процессах, выполняющих буферизацию в адресном пространстве пользователя. Позже процесс может считывать / записывать данные, которые заблокированы / были заблокированы. Функции *stdio* являются наиболее распространенным источником неожиданной буферизации.

Функция *alarm()* может использоваться для обеспечения тайм-аута в приложениях, требующих этого.

Обоснование

Нет.

Будущие направления

lockf - Man Page

См. Также

`сигнализация()`, `chmod()`, `Закреть()`, `creat()`, `fcntl()`, `fopen()`, `mmap()`,
`Открыть()`, `Читать()`, `запись()`

Базовый том определений POSIX.1-2017, `<unistd.h>`

Авторские права

Части этого текста перепечатаны и воспроизведены в электронном виде из 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 and The Open Группа. В случае любого несоответствия между этой версией и исходным стандартом IEEE и Open Group исходный стандарт IEEE и Open Group является документом 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 введены во время преобразования исходных файлов в формат man page. Чтобы сообщить о таких ошибках, см. https://www.kernel.org/doc/man-pages/reporting_bugs.html .

Ссылка на

`mmap (3p)`, `unistd.h(0p)`.

2017 IEEE/The Open Group POSIX Programmer's Manual

lockf - Man Page

[Главная](#) [Блог](#) [0 нас](#)