

fpathconf - Man Page

get configurable pathname variables

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.

Краткий обзор

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

```
long fpathconf(int fildes, int name);  
long pathconf(const char *path, int name);
```

Описание

Функции *fpathconf()* и *pathconf()* определяют текущее значение настраиваемого предела или параметра (*переменной*), связанного с файлом или каталогом.

Для функции *pathconf()* аргумент *path* указывает на путь к файлу или каталогу.

Для *fpathconf()* аргумент *fildes* является открытым файловым дескриптором.

Аргумент *name* представляет переменную, подлежащую запросу относительно этого файла или каталога. Реализации должны поддерживать все

fpathconf - Man Page

`<unistd.h>`, а символические константы, определенные в `<unistd.h>`, являются соответствующими значениями, используемыми для *имени*.

Переменная	Значение <i>имени</i>	Требования
{FILESIZEBITS}	_PC_FILESIZEBITS	4,7
{LINK_MAX}	_PC_LINK_MAX	1
{MAX_CANON}	_PC_MAX_CANON	2
{MAX_INPUT}	_PC_MAX_INPUT	2
{NAME_MAX}	_PC_NAME_MAX	3,4
{PATH_MAX}	_PC_PATH_MAX	4,5
{PIPE_BUF}	_PC_PIPE_BUF	6
{POSIX2_SYMLINKS}	_PC_2_SYMLINKS	4
{POSIX_ALLOC_SIZE_MIN}	_PC_ALLOC_SIZE_MIN	10
{POSIX_REC_INCR_XFER_SIZE}	_PC_REC_INCR_XFER_SIZE	10
{POSIX_REC_MAX_XFER_SIZE}	_PC_REC_MAX_XFER_SIZE	10
{POSIX_REC_MIN_XFER_SIZE}	_PC_REC_MIN_XFER_SIZE	10
{POSIX_REC_XFER_ALIGN}	_PC_REC_XFER_ALIGN	10
{SYMLINK_MAX}	_PC_SYMLINK_MAX	4,9
_POSIX_CHOWN_RESTRICTED	_PC_CHOWN_RESTRICTED	7
_POSIX_NO_TRUNC	_PC_NO_TRUNC	3,4
_POSIX_VDISABLE	_PC_VDISABLE	2
_POSIX_ASYNC_IO	_PC_ASYNC_IO	8
_POSIX_PRIO_IO	_PC_PRIO_IO	8
_POSIX_SYNC_IO	_PC_SYNC_IO	8
_POSIX_TIMESTAMP_RESOLUTION	_PC_TIMESTAMP_RESOLUTION	1

fpathconf - Man Page

1. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно применяться к самому каталогу.
2. Если *path* или *filde*s не ссылаются на файл терминала, не указано, поддерживает ли реализация ассоциацию имени переменной с указанным файлом.
3. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно применяться к именам файлов внутри каталога.
4. Если *path* или *filde*s не ссылаются на каталог, не указано, поддерживает ли реализация ассоциацию имени переменной с указанным файлом.
5. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно быть максимальной длиной относительного пути, который не пересекает точки монтирования, если указанный каталог является рабочим каталогом.
6. Если *path* ссылается на FIFO, или *filde*s ссылается на канал или FIFO, возвращаемое значение должно применяться к объекту, на который ссылается ссылка. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно применяться к любому FIFO, который существует или может быть создан в каталоге. Если *path* или *filde*s ссылаются на какой-либо другой тип файла, не указано, поддерживает ли реализация ассоциацию имени переменной с указанным файлом.
7. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно применяться ко всем файлам, отличным от каталогов, которые существуют или могут быть созданы в каталоге.
8. Если *path* или *filde*s ссылаются на каталог, не указано, поддерживает ли реализация ассоциацию имени переменной с указанным файлом.
9. Если *path* или *filde*s ссылаются на каталог, возвращаемое значение должно быть максимальной длиной строки, которую может содержать символическая ссылка в этом каталоге.

fpathconf - Man Page

файлом. Если реализация поддерживает такую ассоциацию не для обычного файла, возвращаемое значение не указано.

Возвращаемое значение

Если *name* является недопустимым значением, *pathconf()* и *fpathconf()* должны возвращать `-1` и устанавливать *errno* для указания ошибки.

Если переменная, соответствующая *имени*, описана в *<limits.h>* как максимальное или минимальное значение и переменная не имеет ограничений для пути или файлового дескриптора, *pathconf()* и *fpathconf()* должны возвращать `-1` без изменения *errno*. Обратите внимание, что неопределенные пределы не подразумевают бесконечных пределов; см. *<limits.h>*.

Если реализации необходимо использовать *path* для определения значения *name* и реализация не поддерживает ассоциацию *name* с файлом, указанным по *path*, или если процесс не имел соответствующих привилегий для запроса файла, указанного по *path*, или *path* не существует, *pathconf()* возвращает `-1` и установите *errno*, чтобы указать на ошибку.

Если реализация должна использовать *fildes* для определения значения *name* и реализация не поддерживает ассоциацию *name* с файлом, указанным *fildes*, или если *fildes* является недопустимым файловым дескриптором, *fpathconf()* возвращает `-1` и устанавливает *errno* для указания ошибки.

В противном случае *pathconf()* или *fpathconf()* вернут текущее значение переменной для файла или каталога без изменения *errno*. Возвращаемое значение не должно быть более ограничительным, чем соответствующее значение, доступное приложению, когда оно было скомпилировано с помощью *<limits.h>* или *<unistd.h>* реализации.

Если переменная, соответствующая *имени*, зависит от неподдерживаемого параметра, результаты не указаны.

fpathconf - Man Page

Функция *pathconf()* завершится ошибкой, если:

ЭЙНВАЛЬ

Значение *name* недопустимо.

EOVERFLOW

Значение *name* равно `_PC_TIMESTAMP_RESOLUTION`, а разрешение больше `{LONG_MAX}` .

Функция *pathconf()* может завершиться ошибкой, если:

EACCES

Для компонента префикса пути отказано в разрешении поиска.

EINVAL

Реализация не поддерживает ассоциацию имени переменной с указанным файлом.

ЭЛУП

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

ЭЛУП

Во время разрешения аргумента *path* было обнаружено более `{SYMLoop_MAX}` символических ссылок.

ENAMETOOLONG

Длина компонента пути больше `{NAME_MAX}` .

ENAMETOOLONG

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

fpathconf - Man Page

Компонент *path* не называет существующий файл или *путь* является пустой строкой.

ЭНОТДИР

Компонент префикса *path* называет существующий файл, который не является ни каталогом, ни символической ссылкой на каталог, или аргумент *path* содержит по крайней мере один символ, не являющийся `<slash>`, и заканчивается одним или несколькими конечными символами `<slash>`, а последний компонент *pathname* называет существующий файл, который не является ни каталогом, ни символической ссылкой на каталог. каталог или символическая ссылка на каталог.

Функция *fpathconf()* завершится ошибкой, если:

EINVAL

Значение *name* недопустимо.

EOVERFLOW

Значение *name* равно `_PC_TIMESTAMP_RESOLUTION`, а разрешение больше `{LONG_MAX}` .

Функция *fpathconf()* может завершиться ошибкой, если:

EBADF

Аргумент *filde*s не является допустимым файловым дескриптором.

EINVAL

Реализация не поддерживает ассоциацию имени переменной с указанным файлом.

fpathconf - Man Page

Примеры

Нет.

Использование приложений

Разработчики приложений должны проверить, поддерживается ли опция, такая как `_POSIX_ADVISORY_INFO`, до получения и использования значений для связанных переменных, таких как `{POSIX_ALLOC_SIZE_MIN}`.

Обоснование

Функция *pathconf()* была предложена сразу после функции *sysconf()*, когда стало понятно, что некоторые настраиваемые значения могут отличаться в зависимости от файловой системы, каталога или границ устройства.

Например, `{NAME_MAX}` часто меняется между файловыми системами System V и BSD; System V использует максимум 14, BSD 255. В реализации, предоставляющей оба типа файловых систем, приложение будет вынуждено ограничить все компоненты пути 14 байтами, так как это будет значение, указанное в *<limits.h>* в такой системе.

Therefore, various useful values can be queried on any pathname or file descriptor, assuming that appropriate privileges are in place.

Значение, возвращаемое для переменной `{PATH_MAX}`, указывает самый длинный относительный путь, который может быть задан, если указанный каталог является текущим рабочим каталогом процесса. Процесс не всегда может генерировать такое длинное имя и использовать его, если подкаталог в пути переходит в более ограничительную файловую систему. Обратите внимание, что реализации могут принимать имена путей длиннее байтов `{PATH_MAX}`, но не могут возвращать имена путей длиннее этого

fpathconf - Man Page

Значение, возвращаемое для переменной `_POSIX_CHOWN_RESTRICTED`, также применяется к каталогам, в которых не установлены файловые системы. Значение может измениться при пересечении точки монтирования, поэтому приложения, которым это необходимо знать, должны проверять каждый каталог. (Еще более простая проверка – попробовать функцию `chown()` и поискать ошибку в случае, если она произойдет.)

В отличие от значений, возвращаемых `sysconf()`, переменные, ориентированные на путь, потенциально более изменчивы и не гарантированно остаются постоянными на протяжении всего срока службы процесса. Например, между двумя вызовами `pathconf()` рассматриваемая файловая система могла быть размонтирована и перемонтирована с различными характеристиками.

Также обратите внимание, что большинство ошибок являются необязательными. Если одна из переменных всегда имеет одно и то же значение в реализации, реализации не нужно смотреть на *path* или *files*, чтобы вернуть это значение, и поэтому не требуется обнаруживать какие-либо ошибки, кроме значения `[EINVAL]`, которое указывает, что значение *name* недопустимо для этой переменной, и **ошибка** `[EOVERFLOW]`, указывающая на возвращаемое значение, больше `{LONG_MAX}`.

Если значение любого из пределов не определено (логически бесконечно), они не будут определены в `<limits.h>`, а функции `pathconf()` и `fpathconf()` вернут `-1` без изменения `errno`. Это можно отличить от случая предоставления неопознанного аргумента имени, поскольку в этом случае `errno` имеет значение `[EINVAL]`.

Поскольку `-1` является допустимым возвращаемым значением для функций `pathconf()` и `fpathconf()`, приложения должны установить `errno` равным нулю перед их вызовом и проверять `errno` только в том случае, если возвращаемое значение равно `-1`.

fpathconf - Man Page

Символическую ссылку.

Целью IEEE Std 1003.1d-1999 было, чтобы следующие переменные:

```
{POSIX_ALLOC_SIZE_MIN} {POSIX_REC_INCR_XFER_SIZE}  
{POSIX_REC_MAX_XFER_SIZE} {POSIX_REC_MIN_XFER_SIZE}  
{POSIX_REC_XFER_ALIGN}
```

применяется только к обычным файлам, но Примечание 10 также разрешает реализацию консультативной семантики на других типах файлов, уникальных для реализации (например, символьное специальное устройство).

Ошибка **[EOVERFLOW]** для `_PC_TIMESTAMP_RESOLUTION` не может возникнуть в файловых системах, совместимых с POSIX, поскольку POSIX требует разрешения метки времени не более одной секунды. Даже в 32-битных системах это может быть представлено без переполнения.

Будущие направления

Нет.

См. Также

`чаун()`, `конфстр()`, `sysconf()`

Объем базовых определений POSIX.1-2017, `<limits.h>`, `<unistd.h>`

Объем оболочки и утилит POSIX.1-2017, `getconf`

Авторские права

fpathconf - Man Page

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 GroupГруппа. 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 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 .

Ссылка

`chown (3p)`, `confstr (3p)`, `getconf(1p)`, `limits.h(0p)`, `realpath(3p)`, `sysconf (3p)`, `unistd.h(0p)`.

2017 IEEE/The Open Group POSIX Programmer's Manual

fpathconf - Man Page

[Главная](#) [Блог](#) [0 нас](#)