bind(3p) — Linux manual page

PROLOG top

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

НАЗВАНИЕ вверху

bind — привязка имени к сокету

СИНОПСИС сверху

```
#включить <sys/socket.h>

int bind(int socket, const struct sockaddr *address,
socklen_t address_len);
```

ОПИСАНИЕ вверху

Функция *bind*() назначает адрес локального сокета
к сокету, идентифицированному *сокетом дескриптора*, которому не назначен локальный
адрес сокета. Сокеты, созданные с *помощью функции socket*()
, изначально не имеют имен; они идентифицируются только по семейству
адресов.

Функция *bind*() принимает следующие аргументы:

socket Указывает файловый дескриптор сокета, который будет
связан.

адрес указывает на **sockaddr** структура, содержащая адрес
, который будет привязан к сокету. Длина и формат
адреса зависят от семейства адресов
сокета.

address_len Указывает длину структуры **sockaddr**
, на которую указывает аргумент *address*.

Сокет, указанный *сокетом*, может потребовать, чтобы процесс имел
соответствующие привилегии для использования функции *bind*().

Если семейство адресов сокета - AF_UNIX, а путь
в адресе - символическая ссылка, *bind*() завершится ошибкой и установит *errno*
в **[EADDRINUSE]**.

Если адрес сокета не может быть назначен немедленно и
O_NONBLOCK устанавливается для файлового дескриптора сокета, *bind*()
произойдет сбой и установите *errno* в **[EINPROGRESS]**, но
запрос на назначение не будет прерван, и назначение будет
выполнено асинхронно. Последующие вызовы *функции bind*() для одного и того же
сокета до завершения присваивания завершатся ошибкой и установят
errno **равным [EALREADY]**.

Если назначение выполнено асинхронно, *выполните команды pselect*(),

select() и *poll*() указывает, что файловый дескриптор
сокета готов к чтению и записи.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ top

После успешного завершения *bind*() возвращает 0; в противном случае возвращается значение -1
и устанавливается значение errno для указания ошибки.

ОШИБКИ сверху

Функция *bind*() завершится ошибкой, если:

EADDRINUSE
Указанный адрес уже используется.

EADDRNOTAVAIL
Указанный адрес недоступен на локальном
компьютере.

EAFNOSUPPORT
Указанный адрес не является допустимым адресом для семейства
адресов указанного сокета.

EALREADY
Запрос назначения уже выполняется для
указанного сокета.

EBADF Аргумент *сокета* не является допустимым файловым дескриптором.

EINPROGRESS
O_NONBLOCK устанавливается для файлового дескриптора сокета
, и назначение не может быть выполнено немедленно;
назначение должно выполняться асинхронно.

EINVAL Сокет уже привязан к адресу, и
протокол не поддерживает привязку к новому адресу; или
сокет был закрыт.

ENOBUFS
Для завершения
вызова было недостаточно ресурсов.

RUOTSOCK
Аргумент *socket* не ссылается на сокет.

EOPNOTSUPP
Тип сокета указанного сокета не поддерживает
привязку к адресу.

Если семейство адресов сокета AF_UNIX, то *bind*()
завершится ошибкой, если:

НАПРИМЕР, компонент префикса path запрещает разрешение на поиск
или запрошенное имя требует записи в каталог с
режимом, который запрещает разрешение на запись.

EDESTADDRREQ или **EISDIR**
Аргумент *адреса* является нулевым указателем.

EIO Произошла ошибка ввода-вывода.

ELOOP Существует цикл в символических ссылках, встречающихся при
разрешении пути в *адресе*.

ENAMETOOLONG
Длина компонента пути больше
{NAME_MAX}.

УКАЖИТЕ компонент префикса пути в *адресе*

не дает имя существующему файлу или путь к нему является пустой
строкой.

ENOENT или **ENOTDIR**
Путь в *адресе* содержит хотя бы один не-<slash>
символ и заканчивается одним или несколькими завершающими <косыми чертами>
Персонажи. Если путь без завершающей <косой черты>
символы будут называть существующий файл, **ошибка [ENOENT]**
не должна возникать.

ЭНОТДИР
Компонент префикса пути пути в *адресе*
имя существующего файла, который не является ни каталогом, ни
символической ссылкой на каталог, ни путем в *адресе*.
содержит по крайней мере один символ, не являющийся символом <slash>, и заканчивается
одним или несколькими завершающими символами <slash>, а последний
компонент pathname называет существующий файл, который не является ни
каталогом, ни символической ссылкой на каталог.

EROFS Имя будет находиться в файловой системе, доступной только для чтения.

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

EACCES Указанный адрес защищен, и у текущего пользователя
нет разрешения на привязку к нему.

EINVAL The *address_len* аргумент не является допустимой длиной для семейства
адресов.

ЭЙСКОНН
Розетка уже подключена.

При разрешении пути в адресе было обнаружено более {SYMLOOP_MAX} символических ссылок
.

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

Следующие разделы являются информативными.

EXAMPLES top

The following code segment shows how to create a socket and bind it to a name in the AF_UNIX domain.

```
#define MY_SOCK_PATH "/somepath"

int sfd;
struct sockaddr_un my_addr;

sfd = socket(AF_UNIX, SOCK_STREAM, 0);
if (sfd == -1)
    /* Handle error */;

memset(&my_addr, '\0', sizeof(struct sockaddr_un));
                    /* Clear structure */
my_addr.sun_family = AF_UNIX;
strncpy(my_addr.sun_path, MY_SOCK_PATH, sizeof(my_addr.sun_path) -1);

if (bind(sfd, (struct sockaddr *) &my_addr,
    sizeof(struct sockaddr_un)) == -1)
    /* Handle error */;
```

APPLICATION USAGE top

An application program can retrieve the assigned socket name with the *getsockname*() function.

RATIONALE top

None.

FUTURE DIRECTIONS top

None.

SEE ALSO top

connect(3p), getsockname(3p), listen(3p), socket(3p)

The Base Definitions volume of POSIX.1-2017, sys_socket.h(0p)

COPYRIGHT top

Pages that refer to this page: netdb.h(0p), sys_socket.h(0p), sys_un.h(0p), accept(3p), connect(3p), getpeername(3p), getsockname(3p), getsockopt(3p), setsockopt(3p), socket(3p)