

# ассерт(3p) - Страница руководства Linux

[ПРОЛОГ](#) | [Имя](#) | [КРАТКОЕ ОПИСАНИЕ](#) | [Описание](#) | [ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ](#) | [ОШИБКИ](#) | [ПРИМЕРЫ](#) | [ПРИМЕНЕНИЕ](#) | [ОБОСНОВАНИЕ](#) | [БУДУЩИЕ НАПРАВЛЕНИЯ](#) | [СМ. ТАКЖЕ](#) | [Авторские права](#)

## АССЕРТ(3P) POSIX Programmer's Manual АССЕРТ(3P)

### ПРОЛОГ [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 [top](#)

ассерт – принять новое соединение в сокете

### СИНОПСИС [top](#)

#включить <sys/socket.h>

```
int accept(int socket, struct sockaddr *restrict address,
socklen_t *restrict address_len);
```

### ОПИСАНИЕ [top](#)

Функция `accept()` извлекает первое соединение из очереди ожидающих соединений, создает новый сокет с тем же протоколом типа сокета и семейством адресов, что и указанный сокет, и выделяет новый файловый дескриптор для этого сокета. Файловый дескриптор должен быть распределен, как описано в *разделе 2.14*, *Распределение дескрипторов*.

Функция `accept()` принимает следующие аргументы:

`socket` Указывает сокет, который был создан с помощью `socket()`, был привязан к адресу с помощью `bind()`, и выдал успешный вызов `listen()` .

`адресуйте` либо нулевой указатель, либо указатель на `sockaddr` структура, в которой должен быть возвращен адрес соединительного сокета.

`address_len` Либо нулевой указатель, если `адрес` является нулевым указателем, либо указатель на объект, который при вводе указывает длину поставляемого `sockaddr` структура, а на выходе указывает длину сохраненного адреса.

Если `адрес` не является нулевым указателем, адрес однорангового узла для принятого соединения должен храниться в структуре `sockaddr` , на которую указывает `address`, а длина этого адреса должна храниться в объекте, на который указывает `address_len`.

Если фактическая длина адреса больше длины поставляемой структуры, сохраненный адрес должен быть усечен.

Если протокол разрешает соединения несвязанных клиентов, а одноранговый узел не связан, то значение, хранящееся в объекте, на который указывает `адрес` не указан.

Если очередь прослушивания пуста от запросов на подключение и `O_NONBLOCK` не установлен в файловом дескрипторе сокета, `accept()` блокируется до тех пор, пока не будет установлено соединение. Если в очереди `listen()` нет запросов на подключение, а в файловом дескрипторе сокета установлен `O_NONBLOCK`, `accept()` завершится ошибкой и установит `errno` в `[EAGAIN]` или `[EWOULDBLOCK]`.

Принятый сокет не может сам принимать больше соединений. Исходный сокет остается открытым и может принимать больше соединений.

## ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ [top](#)

После успешного завершения `accept()` возвращает неотрицательный файловый дескриптор принятого сокета. В противном случае возвращается значение `-1`, `errno` указывает на ошибку, и любой объект, на который указывает `address_len`, остается неизменным.

## ОШИБКИ [сверху](#)

Функция `accept()` завершится ошибкой, если:

### **EAGAIN** или **EWOULDBLOCK**

`O_NONBLOCK` устанавливается для дескриптора файла сокета, и никакие соединения не принимаются.

**EBADF** Аргумент `сокета` не является допустимым файловым дескриптором.

### **ECONNABORTED**

Соединение прервано.

Функция **EINTR** `accept()` была прервана сигналом, который был пойман до прибытия действительного соединения.

*Сокет **EINVAL*** не принимает соединения.

**EMFILE** Все файловые дескрипторы, доступные процессу, в настоящее время открыты.

**ENFILE** Максимальное количество файловых дескрипторов в системе уже открыто.

### **ENOBUFFS**

Буферное пространство отсутствует.

**ENOMEM** Для завершения операции не хватило памяти

.

### **RUOTSOCK**

Аргумент *socket* не ссылается на сокет.

### **EOPNOTSUPP**

Тип сокета указанного сокета не поддерживает прием соединений.

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

**EPROTO** Произошла ошибка протокола; например, ПОТОКИ стек протоколов не инициализирован.

*Следующие разделы являются информативными.*

## **ПРИМЕРЫ** [сверху](#)

Нет.

## **ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЙ** [сверху](#)

Если соединение доступно, *select()* указывает, что файловый дескриптор сокета готов для чтения.

## RATIONALE [top](#)

None.

## FUTURE DIRECTIONS [top](#)

None.

## СМ. ТАКЖЕ [top](#)

*Раздел 2.14, Распределение файловых дескрипторов*, [bind\(3p\)](#), [connect\(3p\)](#), [listen\(3p\)](#), [socket \(3p\)](#)

Базовый том определений POSIX.1–2017, [sys\\_socket.h\(0p\)](#)

## COPYRIGHT [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 несоответствия 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](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

## IEEE/The Open Group 2017 ACCEPT(3P)

---

Страницы, которые ссылаются на эту страницу: [sys\\_socket.h\(0p\)](#), [connect\(3p\)](#), [getpeername\(3p\)](#), [getsockname\(3p\)](#), [listen\(3p\)](#), [pselect\(3p\)](#), [socket\(3p\)](#)

---

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

For details of in-depth **Linux/UNIX system programming training courses** that I teach, look [here](#).

Hosting by [jambit GmbH](#).

