

NAME

shutdown - disable sends and/or receives on a socket

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
```

int

```
shutdown(int s, int how);
```

DESCRIPTION

The **shutdown()** system call disables sends or receives on a socket. The *how* argument specifies the type of shutdown. Possible values are:

SHUT_RD Further receives will be disallowed.

SHUT_WR Further sends will be disallowed. This may cause actions specific to the protocol family of the socket *s* to happen; see *IMPLEMENTATION NOTES*.

SHUT_RDWR Further sends and receives will be disallowed. Implies SHUT_WR.

If the file descriptor *s* is associated with a SOCK_STREAM socket, all or part of the full-duplex connection will be shut down.

IMPLEMENTATION NOTES

The following protocol specific actions apply to the use of SHUT_WR (and potentially also SHUT_RDWR), based on the properties of the socket associated with the file descriptor *s*.

Domain	Type	Protocol	Action
PF_INET	SOCK_DGRAM	IPPROTO_SCTP	Failure, as socket is not connected.
PF_INET	SOCK_DGRAM	IPPROTO_UDP	Failure, as socket is not connected.
PF_INET	SOCK_STREAM	IPPROTO_SCTP	Send queued data and tear down association.
PF_INET	SOCK_STREAM	IPPROTO_TCP	Send queued data, wait for ACK, then send FIN.
PF_INET6	SOCK_DGRAM	IPPROTO_SCTP	Failure, as socket is not connected.
PF_INET6	SOCK_DGRAM	IPPROTO_UDP	Failure, as socket is not connected.

PF_INET6	SOCK_STREAM	IPPROTO_SCTP	Send queued data and tear down association.
PF_INET6	SOCK_STREAM	IPPROTO_TCP	Send queued data, wait for ACK, then send FIN.

RETURN VALUES

The **shutdown()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **shutdown()** system call fails if:

[EBADF]	The <i>s</i> argument is not a valid file descriptor.
[EINVAL]	The <i>how</i> argument is invalid.
[ENOTCONN]	The <i>s</i> argument specifies a socket which is not connected.
[ENOTSOCK]	The <i>s</i> argument does not refer to a socket.

SEE ALSO

connect(2), socket(2), inet(4), inet6(4)

STANDARDS

The **shutdown()** system call is expected to comply with IEEE Std 1003.1g-2000 ("POSIX.1g"), when finalized.

HISTORY

The **shutdown()** system call appeared in 4.2BSD. The SHUT_RD, SHUT_WR, and SHUT_RDWR constants appeared in IEEE Std 1003.1g-2000 ("POSIX.1g").

AUTHORS

This manual page was updated by Bruce M. Simpson <bms@FreeBSD.org> to reflect how **shutdown()** behaves with PF_INET and PF_INET6 sockets.

BUGS

The ICMP "port unreachable" message should be generated in response to datagrams received on a local port to which *s* is bound after **shutdown()** is called.