

NAME

socketpair – create a pair of connected sockets

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

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

```
int socketpair(int domain, int type, int protocol, int sv[2]);
```

DESCRIPTION

The **socketpair()** call creates an unnamed pair of connected sockets in the specified *domain*, of the specified *type*, and using the optionally specified *protocol*. For further details of these arguments, see **socket(2)**.

The file descriptors used in referencing the new sockets are returned in *sv[0]* and *sv[1]*. The two sockets are indistinguishable.

RETURN VALUE

On success, zero is returned. On error, *-1* is returned, *errno* is set to indicate the error, and *sv* is left unchanged.

On Linux (and other systems), **socketpair()** does not modify *sv* on failure. A requirement standardizing this behavior was added in POSIX.1-2008 TC2.

ERRORS**EAFNOSUPPORT**

The specified address family is not supported on this machine.

EFAULT

The address *sv* does not specify a valid part of the process address space.

EMFILE

The per-process limit on the number of open file descriptors has been reached.

ENFILE

The system-wide limit on the total number of open files has been reached.

EOPNOTSUPP

The specified protocol does not support creation of socket pairs.

EPROTONOSUPPORT

The specified protocol is not supported on this machine.

STANDARDS

POSIX.1-2001, POSIX.1-2008, 4.4BSD. **socketpair()** first appeared in 4.2BSD. It is generally portable to/from non-BSD systems supporting clones of the BSD socket layer (including System V variants).

NOTES

On Linux, the only supported domains for this call are **AF_UNIX** (or synonymously, **AF_LOCAL**) and **AF_TIPC** (since Linux 4.12).

Since Linux 2.6.27, **socketpair()** supports the **SOCK_NONBLOCK** and **SOCK_CLOEXEC** flags in the *type* argument, as described in **socket(2)**.

SEE ALSO

pipe(2), **read(2)**, **socket(2)**, **write(2)**, **socket(7)**, **unix(7)**