

**NAME**

socketpair – create a pair of connected sockets

**SYNOPSIS**

```
#include <sys/types.h>      /* See NOTES */
#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 appropriately, 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.1-2016.

**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.

**CONFORMING TO**

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)**.

POSIX.1 does not require the inclusion of *<sys/types.h>*, and this header file is not required on Linux. However, some historical (BSD) implementations required this header file, and portable applications are probably wise to include it.

**SEE ALSO**

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

**COLOPHON**

This page is part of release 5.05 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.