

[Skip site navigation](#) (1) [Skip section navigation](#) (2)

Site Navigation

FreeBSD Manual Pages

select

man

apropos

2 - System Calls

▼ FreeBSD 3.3-RELEASE

▼

All Architectures ▼ html ▼

[home](#) | [help](#)

SELECT(2)

System Calls Manual

SELECT(2)

NAME

select -- synchronous I/O multiplexing

SYNOPSIS

```
#include <sys/types.h>
#include <sys/time.h>
#include <unistd.h>
```

```
int
select(int nfd, fd_set *readfds, fd_set *writefds, fd_set *exceptfds,
       struct timeval *timeout);
```

```
FD_SET(fd, &fdset);
```

```
FD_CLR(fd, &fdset);
```

```
FD_ISSET(fd, &fdset);
```

```
FD_ZERO(&fdset);
```

DESCRIPTION

Select() examines the I/O descriptor sets whose addresses are passed in *readfds*, *writefds*, and *exceptfds* to see if some of their descriptors are ready for reading, are ready for writing, or have an exceptional condition pending, respectively. The only exceptional condition detectable is out-of-band data received on a socket. The first *nfd* descriptors are checked in each set; i.e., the descriptors from 0 through *nfd*-1 in the descriptor sets are examined. On return, **select()** replaces the given descriptor sets with subsets consisting of those descriptors that are ready for the requested operation. **Select()** returns the total number of ready descriptors in all the sets.

The descriptor sets are stored as bit fields in arrays of integers. The following macros are provided for manipulating such descriptor sets: **FD_ZERO(&fdset)** initializes a descriptor set *fdset* to the null set. **FD_SET(fd, &fdset)** includes a particular descriptor *fd* in *fdset*. **FD_CLR(fd, &fdset)** removes *fd* from *fdset*. **FD_ISSET(fd, &fdset)** is non-zero if *fd* is a member of *fdset*, zero otherwise. The behavior of these macros is undefined if a descriptor value is less than zero or greater

Amazon Ex. 1023
IPR Petition - USP 7,784,058

than or equal to `FD_SETSIZE`, which is normally at least equal to the maximum number of descriptors supported by the system.

If *timeout* is a non-nil pointer, it specifies a maximum interval to wait for the selection to complete. If *timeout* is a nil pointer, the select blocks indefinitely. To effect a poll, the *timeout* argument should be non-nil, pointing to a zero-valued `timeval` structure.

Any of *readfds*, *writefds*, and *exceptfds* may be given as nil pointers if no descriptors are of interest.

RETURN VALUES

Select() returns the number of ready descriptors that are contained in the descriptor sets, or -1 if an error occurred. If the time limit expires, **select()** returns 0. If **select()** returns with an error, including one due to an interrupted call, the descriptor sets will be unmodified.

ERRORS

An error return from **select()** indicates:

[EBADF]	One of the descriptor sets specified an invalid descriptor.
[EINTR]	A signal was delivered before the time limit expired and before any of the selected events occurred.
[EINVAL]	The specified time limit is invalid. One of its components is negative or too large.
[EINVAL]	<i>nfds</i> was invalid.

SEE ALSO

[accept\(2\)](#), [connect\(2\)](#), [getdtablesize\(2\)](#), [gettimeofday\(2\)](#), [read\(2\)](#), [recv\(2\)](#), [send\(2\)](#), [write\(2\)](#), [clocks\(7\)](#)

NOTES

The default size of `FD_SETSIZE` is currently 1024. In order to accommodate programs which might potentially use a larger number of open files with **select()**, it is possible to increase this size by having the program define `FD_SETSIZE` before the inclusion of any header which includes `<sys/types.h>`.

If *nfds* is greater than the number of open files, **select()** is not guaranteed to examine the unused file descriptors. For historical reasons, **select()** will always examine the first 256 descriptors.

BUGS

select() should probably return the time remaining from the original timeout, if any, by modifying the time value in place. This may be implemented in future versions of the system. Thus, it is unwise to assume that the timeout value will be unmodified by the **select()** call.

HISTORY

The **select()** function call appeared in 4.2BSD.

4.2 Berkeley Distribution

March 25, 1994

SELECT(2)

[NAME](#) | [SYNOPSIS](#) | [DESCRIPTION](#) | [RETURN VALUES](#) | [ERRORS](#) | [SEE ALSO](#) | [NOTES](#) | [BUGS](#) | [HISTORY](#)

Want to link to this manual page? Use this URL:

<<https://man.freebsd.org/cgi/man.cgi?query=select&sektion=2&manpath=FreeBSD+3.3-RELEASE>>

[home](#) | [help](#)

[Legal Notices](#) | © 1995-2025 The FreeBSD Project. All rights reserved.

[Contact](#)