NAME

poll – input/output multiplexing

SYNOPSIS

#include <stropts.h>
#include <poll.h>

int poll(struct pollfd **fds*, **unsigned long** *nfds*, **int** *timeout***);**

DESCRIPTION

poll( ) provides users with a mechanism for multiplexing input/output over a set of file descriptors that reference open files. **poll( )** identifies those files on which a user can send or receive messages, or on which certain events have occurred.

fds specifies the file descriptors to be examined and the events of interest for each file descriptor. It is a pointer to an array with one element for each open file descriptor of interest. The array's elements are **pollfd** structures, which contain the following members:

```
int fd;             /* file descriptor */
short events;       /* requested events */
short revents;      /* returned events */
```

fd specifies an open file descriptor and **events** and **revents** are bitmasks constructed by an OR of any combination of the following event flags:

POLLIN Data other than high priority data may be read without blocking. For STREAMS, this flag is set even if the message is of zero length.

POLLRDNORM Normal data (priority band = 0) may be read without blocking. For STREAMS, this flag is set even if the message is of zero length.

POLLRDBAND Data from a non-zero priority band may be read without blocking For STREAMS, this flag is set even if the message is of zero length.

POLLPRI High priority data may be received without blocking. For STREAMS, this flag is set even if the message is of zero length.

POLLOUT Normal data may be written without blocking.

POLLWRNORM The same as **POLLOUT**.

POLLWRBAND Priority data (priority band > 0) may be written. This event only examines bands that have been written to at least once.

POLLERR An error has occurred on the device or stream. This flag is only valid in the **revents** bitmask; it is not used in the **events** field.

POLLHUP A hangup has occurred on the stream. This event and **POLLOUT** are mutually exclusive; a stream can never be writable if a hangup has occurred. However, this event and **POLLIN**, **POLLRDNORM**, **POLLRDBAND**, or **POLLPRI** are not mutually exclusive. This flag is only valid in the **revents** bitmask; it is not used in the **events** field.

POLLNVAL The specified **fd** value does not belong to an open file. This flag is only valid in the **revents** field; it is not used in the **events** field.

For each element of the array pointed to by *fds*, **poll( )** examines the given file descriptor for the event(s) specified in **events**. The number of file descriptors to be examined is specified by *nfds*.

If the value **fd** is less than zero, **events** is ignored and **revents** is set to 0 in that entry on return from **poll( )**.

The results of the **poll( )** query are stored in the **revents** field in the **pollfd** structure. Bits are set in the **revents** bitmask to indicate which of the requested events are true. If none are true, none of the specified bits are set in **revents** when the **poll( )** call returns. The event flags **POLLHUP**, **POLLERR**, and **POLLNVAL** are always set in **revents** if the conditions they indicate are true; this occurs even though these flags were not present in **events**.

If none of the defined events have occurred on any selected file descriptor, **poll( )** waits at least *timeout* milliseconds for an event to occur on any of the selected file descriptors. On a computer where millisecond timing accuracy is not available, *timeout* is rounded up to the nearest legal value available on that system. If the value *timeout* is 0, **poll( )** returns immediately. If the value of *timeout* is **INFTIM** (or −1), **poll( )** blocks until a requested event occurs or until the call is interrupted. **poll( )** is not affected by the **O_NDELAY** and **O_NONBLOCK** flags.

RETURN VALUES

Upon successful completion, a non-negative value is returned. A positive value indicates the total number of file descriptors that has been selected (that is, file descriptors for which the **revents** field is non-zero). A value of 0 indicates that the call timed out and no file descriptors have been selected. Upon failure, a value of −1 is returned and **errno** is set to indicate the error.

ERRORS

poll( ) fails if one or more of the following are true:

EAGAIN Allocation of internal data structures failed, but the request may be attempted again.

EFAULT Some argument points to an illegal address.

EINTR A signal was caught during the **poll()** function.

EINVAL The argument *nfds* is greater than **{OPEN_MAX}**.

SEE ALSO

intro(2), **getmsg**(2), **getrlimit**(2), **putmsg**(2), **read**(2), **write**(2), **chpoll**(9E)

STREAMS Programming Guide

NOTES

Non-STREAMS drivers use **chpoll**(9E) to implement **poll** on these devises.