

NAME

socketmark – determine whether socket is at out-of-band mark

SYNOPSIS

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

```
int socketmark(int sockfd);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros(7)**):

```
socketmark(): _POSIX_C_SOURCE >= 200112L
```

DESCRIPTION

socketmark() returns a value indicating whether or not the socket referred to by the file descriptor *sockfd* is at the out-of-band mark. If the socket is at the mark, then 1 is returned; if the socket is not at the mark, 0 is returned. This function does not remove the out-of-band mark.

RETURN VALUE

A successful call to **socketmark()** returns 1 if the socket is at the out-of-band mark, or 0 if it is not. On error, -1 is returned and *errno* is set to indicate the error.

ERRORS**EBADF**

sockfd is not a valid file descriptor.

EINVAL

sockfd is not a file descriptor to which **socketmark()** can be applied.

VERSIONS

socketmark() was added to glibc in version 2.2.4.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
socketmark()	Thread safety	MT-Safe

CONFORMING TO

POSIX.1-2001, POSIX.1-2008.

NOTES

If **socketmark()** returns 1, then the out-of-band data can be read using the **MSG_OOB** flag of **recv(2)**.

Out-of-band data is supported only on some stream socket protocols.

socketmark() can safely be called from a handler for the **SIGURG** signal.

socketmark() is implemented using the **SIOCATMARK ioctl(2)** operation.

BUGS

Prior to glibc 2.4, **socketmark()** did not work.

EXAMPLE

The following code can be used after receipt of a **SIGURG** signal to read (and discard) all data up to the mark, and then read the byte of data at the mark:

```
char buf[BUF_LEN];
char oobdata;
int atmark, s;

for (;;) {
    atmark = socketmark(sockfd);
    if (atmark == -1) {
        perror("socketmark");
        break;
    }
}
```

```
        if (atmark)
            break;

        s = read(sockfd, buf, BUF_LEN);
        if (s == -1)
            perror("read");
        if (s <= 0)
            break;
    }

    if (atmark == 1) {
        if (recv(sockfd, &oobdata, 1, MSG_OOB) == -1) {
            perror("recv");
            ...
        }
    }
}
```

SEE ALSO

fcntl(2), **recv(2)**, **send(2)**, **tcp(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/>.