Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sockatmark.3p' command

$ man sockatmark.3p

SOCKATMARK(3P) POSIX Programmer's Manual SOCKATMARK(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

sockatmark ? determine whether a socket is at the out-of-band mark

SYNOPSIS

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

int sockatmark(int s);
```

DESCRIPTION

The sockatmark() function shall determine whether the socket specified by the descriptor s is at the out-of-band data mark (see Section 2.10.12, Socket Out-of-Band Data State). If the protocol for the socket supports out-of-band data by marking the stream with an out-of-band data mark, the sockatmark() function shall return 1 when all data preceding the mark has been read and the out-of-band data mark is the first element in the receive queue. The sockatmark() function shall not remove the mark from the stream.

RETURN VALUE

Upon successful completion, the sockatmark() function shall return a value indicating whether the socket is at an out-of-band data mark. If

the protocol has marked the data stream and all data preceding the mark has been read, the return value shall be 1; if there is no mark, or if data precedes the mark in the receive queue, the sockatmark() function shall return 0. Otherwise, it shall return a value of -1 and set errno to indicate the error.

ERRORS

The sockatmark() function shall fail if:

EBADF The s argument is not a valid file descriptor.

ENOTTY The file associated with the s argument is not a socket.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The use of this function between receive operations allows an applica? tion to determine which received data precedes the out-of-band data and which follows the out-of-band data.

There is an inherent race condition in the use of this function. On an empty receive queue, the current read of the location might well be at the ``mark'', but the system has no way of knowing that the next data segment that will arrive from the network will carry the mark, and sockatmark() will return false, and the next read operation will silently consume the mark.

Hence, this function can only be used reliably when the application al? ready knows that the out-of-band data has been seen by the system or that it is known that there is data waiting to be read at the socket (via SIGURG or select()). See Section 2.10.11, Socket Receive Queue, Section 2.10.12, Socket Out-of-Band Data State, Section 2.10.14, Sig? nals, and pselect() for details.

RATIONALE

The sockatmark() function replaces the historical SIOCATMARK command to ioctl() which implemented the same functionality on many implementa? tions. Using a wrapper function follows the adopted conventions to avoid specifying commands to the ioctl() function, other than those now

included to support XSI STREAMS. The sockatmark() function could be im?

plemented as follows:

```
#include <sys/ioctl.h>
int sockatmark(int s)
{
    int val;
    if (ioctl(s,SIOCATMARK,&val)==-1)
        return(-1);
    return(val);
}
```

The use of [ENOTTY] to indicate an incorrect descriptor type matches the historical behavior of SIOCATMARK.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.10.12, Socket Out-of-Band Data State, pselect(), recv(), recvmsg()

The Base Definitions volume of POSIX.1?2017, <sys_socket.h>

COPYRIGHT