LibNoWebsocket

nowsread, nowswrite, nowsclose — read, write, close a regular stream socket when peer is a websocket.

PROLOG

This manual page will never be part of the POSIX Programmer's Manual.

nowsread, nowswrite and nowsclose are designed to be similar to posix read, write and close. Their purpose is to provide native sockets to HTML5 web browsers. This is not yet allowed by standards, but the other way around is not forbidden. These functions mimic posix's read, write and close and transparently emulate the websocket protocol.

These functions allow their users to communicate with HTML5-enabled web browsers using legacy knowledge and habits gained through years of experience on network programming with TCP stream sockets and POSIX API as a reference.

SYNOPSYS

#include <nows.h>

ssize_t nowsread (int **sock**, void* **data**, size_t **len**);

ssize_t nowswrite (int **sock**, const void* **data**, size_t **len**);

int nowsclose (int **sock**);

int nows_simulate_client (int **sock**);

DESCRIPTION

nowsread(3) is very similar to read(3POSIX) in its arguments, behavior, return value, in blocking or non blocking mode, when **sock** is a STREAMS socket in *byte-stream* mode (*message-nondiscard* and *message-discard* modes are yet unconsidered). Same goes to nowswrite(3) regarding write(3POSIX). The websocket protocol includes a handshake process which sadly breaks symmetry between peers. As generally HTML5 browsers act as websocket clients, nowsread(3) and nowswrite(3) act by default as websocket server (even if the socket **sock** is a SOCK_STREAM client). A call to nows_simulate_client(3) prior to nowsread(3) and nowswrite(3) will switch this behavior.

Nothing beeing perfect, here is a list of specific differences or relevant notices:

blocking mode

> read(3POSIX) blocks the caller until exactly **len** bytes are received. Similarly nowsread(3) blocks the caller until at least one byte is received, but due to the nature of the websocket protocol, this does not guarantee that exactly **len** bytes are received. The returned value indicates the amount of received bytes in the **data** buffer. nowswrite(3) has however the same behavior as write(3POSIX).

non-blocking mode

> Any value (greater than 0) given to **len** will follow read/write(3POSIX) behavior.

websocket specifics or restrictions

> Underlying ping-pong facility is not implemented yet. Only text buffers are sent or received. This is subject for improvements.

nowsclose(3) is similar to close(3POSIX).

nows_simulate_client(3) sets this side of the stream to act as a websocket client.

RETURN VALUE

Please refer to read(3POSIX), write(3POSIX). To summarize, 0 always indicates EOF, -1 an error and in that case **errno** is set. **errno** can be set to EAGAIN even when in blocking mode. However in blocking mode EAGAIN is never continuously returned and cannot induce an active loop. Otherwise the returned value indicates the number of bytes sent or received.

now_simulate_client(3) returns 0 on success, -1 on error and in that case **errno** is set.

ERRORS

They are the same as with read(3POSIX), write(3POSIX) and close(3POSIX). Additionnaly,

ENOMEM
can be returned in case of memory shortage.

EAGAIN
can be returned by nowsread(3) or nowswrite(3) in blocking mode (however never continuously thus not inducing an active loop). This is subject for improvements.

LICENSE

GPLv2 (due to sha1 embedded library - subject for changes)

AUTHOR

CNRS

https://redmine.laas.fr/projects/libnowebsocket