

Ruby trunk - Feature #11229

[PATCH] socket: allow exception-free nonblocking sendmsg/recvmsg

06/05/2015 11:47 PM - normalperson (Eric Wong)

Status:	Closed
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	
Description	
<p>As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.</p> <p>Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.</p> <p>I think I have only one more change I'd like to add for recv/recvmsg* APIs in the future: destination buffers to reduce GC overhead</p>	

Associated revisions

Revision b9a91334 - 06/15/2015 07:38 PM - normal

socket: allow exception-free nonblocking sendmsg/recvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_recvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_recvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_recvfrom_nonblock): use rsock_opt_false_p
- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_recv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_recvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50910 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50910 - 06/15/2015 07:38 PM - normalperson (Eric Wong)

socket: allow exception-free nonblocking sendmsg/recvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_recvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_recvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_recvfrom_nonblock): use rsock_opt_false_p

- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_rcv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_rcvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

Revision 50910 - 06/15/2015 07:38 PM - normal

socket: allow exception-free nonblocking sendmsg/rcvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_rcvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): use rsock_opt_false_p
- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_rcv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_rcvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

Revision 50910 - 06/15/2015 07:38 PM - normal

socket: allow exception-free nonblocking sendmsg/rcvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_rcvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): use rsock_opt_false_p
- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_rcv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_rcvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

Revision 50910 - 06/15/2015 07:38 PM - normal

socket: allow exception-free nonblocking sendmsg/rcvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_rcvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): use rsock_opt_false_p

- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_rcv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_rcvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

Revision 50910 - 06/15/2015 07:38 PM - normal

socket: allow exception-free nonblocking sendmsg/recvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_recvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_rcvfrom_nonblock): use rsock_opt_false_p
- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_rcv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_rcvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

[ruby-core:69542] [Feature #11229]

Revision d51065b0 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50913 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50913 - 06/15/2015 09:28 PM - normalperson (Eric Wong)

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 55a692bb - 11/17/2015 02:29 AM - normal

socket: update doc for rcvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52608 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52608 - 11/17/2015 02:29 AM - normalperson (Eric Wong)

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

History

#1 - 06/12/2015 08:52 AM - matz (Yukihiro Matsumoto)

Looks good to me. Go ahead.

Matz.

#2 - 06/15/2015 07:39 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset [r50910](#).

socket: allow exception-free nonblocking sendmsg/recvmsg

As documented before, exceptions are expensive and IO::Wait*able are too common in socket applications to be the exceptional case. Datagram sockets deserve the same API which stream sockets are allowed with read_nonblock and write_nonblock.

Note: this does not offer a performance advantage under optimal conditions when both ends are equally matched in speed, but it it does make debug output cleaner by avoiding exceptions whenever the receiver slows down.

- ext/socket/ancdata.c (bsock_sendmsg_internal, bsock_recvmsg_internal): support "exception: false" kwarg
- ext/socket/init.c (rsock_s_recvfrom_nonblock): ditto
- ext/socket/init.c (rsock_s_recvfrom_nonblock): use rsock_opt_false_p
- ext/socket/socket.c (sock_connect_nonblock): ditto
- ext/socket/rubysocket.h (rsock_opt_false_p): new function
- ext/socket/basicsocket.c (bsock_recv_nonblock): update rdoc
- ext/socket/udpsocket.c (udp_recvfrom_nonblock): ditto
- test/socket/test_nonblock.rb: new tests

Files

0001-socket-allow-exception-free-nonblocking-sendmsg-recv.patch	13.4 KB	06/05/2015	normalperson (Eric Wong)
---	---------	------------	--------------------------