Ruby trunk - Misc #14760

cross-thread IO#close semantics

05/15/2018 10:04 AM - normalperson (Eric Wong)

Status:	Open	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	

Description

I wrote about cross-thread IO#close in ruby-core, but I'm not sure if it's a bug or not to have missing support for IO.select and IO.copy_stream:

IO.select -
http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/86655
https://public-inbox.org/ruby-core/20180423133946.GA6019@dcvr/

IO.copy_stream -
http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/87040
https://public-inbox.org/ruby-core/20180515095315.GA15909@dcvr/

I know the IO.select case in 1.9+ differs from 1.8, and IO.copy_stream wasn't in 1.8, but I guess the current behavior is that it isn't consistent with normal IO methods. IO.copy_stream will also behave "normally" and raise IOError if it somehow hits non-optimized cases and ends up calling Ruby methods, but my example in [ruby-core:87040] did not hit that case.

On one hand, I'm not a fan of "nanny features" like deadlock detection for threading. On the other hand, I value consistency and we already went down the rabbit hole of supporting current users of rb_thread_io_blocking_region.

Anyways, I can implement these if desired since I have additional work planned in this area anyways (auto-fiber).

History

#1 - 05/17/2018 08:21 AM - akr (Akira Tanaka)

I feel cross-thread IO#close is not reliable way to interrupt other threads.
So, I implemented IO.copy_stream without concerning cross-thread IO#close.

Since consistency is not a goal of Ruby, I think current behavior is acceptable.