Ruby - Bug #13920

prepping io for stdin returns "access denied" on windows

09/19/2017 10:10 AM - rocifier (Ryan O'Connor)

Status:	Closed			
Priority:	Normal			
Assignee:				
Target version:				
ruby -v:	2.5.0		**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description
https://github.com/ruby/ruby/blob/c08f7b80889b531865e74bc5f573df8fa27f2088/io.c#L12730   rb_stdin = prep_stdio(stdin, FMODE_READABLE, rb_cIO, "<STDIN>");   When this line (io.c) is called upon initialising ruby with ruby_init() or ruby_setup() and the library is built with VS2015 x86 Native Tools Command Prompt, fileno(stdin) returns -2 and errno is set to 13. This raises a ruby exception and fails setup. The library has been tested with /MD and /MDd and with 2.4.2 and 2.5.0

History

#1 - 09/19/2017 10:15 AM - rocifier (Ryan O'Connor)

I am linking to the static library from a win32 application (not console app) set to use the same runtime version as the ruby build. I have tried setting up a console with AllocConsole() and assigning *stdin to this console before initializing ruby, but it doesn't help.

#2 - 09/19/2017 12:35 PM - MSP-Greg (Greg L)

rocifier (Ryan O'Connor) wrote:

> fileno(stdin) returns -2

Sounds similar to #13546

#3 - 09/20/2017 07:49 AM - rocifier (Ryan O'Connor)

MSP-Greg (Greg L) wrote:

> rocifier (Ryan O'Connor) wrote:
>
>> fileno(stdin) returns -2
>
> Sounds similar to #13546

Thanks my issue is very similar. The issue you linked to presents some undocumented requirements. When I try calling rb_w32_sysinit it crashes on this line:

https://github.com/ruby/ruby/blob/c08f7b80889b531865e74bc5f573df8fa27f2088/win32/win32.c#L839

What is the exact version of ucrtbased.dll that the assembly code in this function is designed for?

EDIT

I got it working! Basically I had to build with these flags in Makefile for debug:

RUNTIMEFLAG = /MDd LDFLAGS = /NODEFAULTLIB:MSVCRT DEBUGFLAGS = /Zi /D "_DEBUG"

I deleted all traces of the ucrtbased.dll on my system which were of version 10.0.15063.137 and copied version 10.0.10240.16384 to C:\Windows\SysWOW64 (I'm on Windows 10). The correct working version is the one from the Windows 10 SDK.

#4 - 10/17/2019 11:58 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed