Ruby master - Bug #6790

mingw-w64: rb_libruby_handle and libruby changes when extensions are loaded

07/25/2012 08:24 AM - luislavena (Luis Lavena)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>nobu (Nobuyoshi Nakada)</td>
</tr>
<tr>
<td>Target version</td>
<td>2.0.0</td>
</tr>
<tr>
<td>Backport:</td>
<td></td>
</tr>
</tbody>
</table>

ruby -v:
ruby 2.0.0dev (2012-07-25 trunk 36527) [x64-mingw32]

Description

Hello,

I started to look into this failure when running dl/handle tests:

test_initialize_noargs(DL::TestHandle):
DL::DLError: unknown symbol "rb_str_new"
C:/Users/Worker/Jenkins/workspace/git-ruby-trunk/test/dl/test_handle.rb:110:in []'
C:/Users/Worker/Jenkins/workspace/git-ruby-trunk/test/dl/test_handle.rb:110:intest_initialize_noargs'

Above error only happens with compiled 64bits ruby.

After looking a bit more closer, it seems DllMain from libruby gets called several times when (((x64-msvcr7.1-ruby200.dll))) is loaded but also when extensions are loaded.

Adding some debug strings into DllMain like the following:

diff --git a/ruby.c b/ruby.c
--- a/ruby.c
+++ b/ruby.c
@@ -315,8 +315,22 @@ static HMODULE libruby;
     BOOL WINAPI DllMain(HINSTANCE dll, DWORD reason, LPVOID reserved)
     {
         HMODULE self;
     char filename[MAX_PATH]; +   if (reason == DLL_PROCESS_ATTACH)
         libruby = dll;
     { +   self = GetModuleHandle(NULL);
         printf("DLL_PROCESS_ATTACH\n"); +   GetModuleFileName(self, filename, MAX_PATH);
         printf("self: %s\n", self, filename); +   GetModuleFileName(dll, filename, MAX_PATH);
         printf("dll: %s\n", dll, filename); +   libruby = dll;
         
     return TRUE;
     }

Produces the following:

32bits compilation:

C:/Users/Luis/Code/ruby/ruby/x86-installed>ruby -v -rdl -e "puts :ok"
DLL_PROCESS_ATTACH
self: 0x40000000 (C:/Users/Luis/Code/ruby/ruby/x86-installed/bin/ruby.exe)
dll: 0x67800000 (C:/Users/Luis/Code/ruby/ruby/x86-installed/bin/msvcr7.1.dll)
ruby 2.0.0dev (2012-07-25 trunk 36527) [i386-mingw32]
ok

64bits (mingw-w64):
C:\Users\Luis\Code\ruby\ruby\x64-installed>ruby -v -rdl -e "puts :ok"

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x6f100000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\x64-msvcr70-ruby200.dll)
ruby 2.0.0dev (2012-07-25 trunk 36527) [x86-mingw32]

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x1ee00000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\enc\encdb.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x6aa00000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\enc\iso_8859_1.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x68080000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\enc\trans\transdb.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x623c0000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\dll.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x63d80000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\fiddle.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x6e6c0000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\enc\utf_16le.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x6a340000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\lib\ruby\2.0.0\x64-mingw32\enc\single_byte.so)

DLL_PROCESS_ATTACH
self: 0x400000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\ruby.exe)
dll: 0x6f100000 (C:\Users\Luis\Code\ruby\ruby\x64-installed\bin\x64-msvcr70-ruby200.dll)

I just tested a simple project which loads multiple DLLs and DllMain of each DLL do not get invoked several times.

Since each extension depends on (\{x86-msvcr70-ruby200.dll\}), perhaps that is the reason why (\{DllMain\}) is invoked multiple times.

Maybe there is something I'm missing? Perhaps a workaround will be set (\{libruby\}) only once?

=end

Associated revisions
Revision 3f539a5b - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36544 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.
Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 36544 - 07/27/2012 03:14 AM - usa (Usaku NAKAMURA)

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

Revision 6249dac0 - 09/25/2012 12:30 AM - naruse (Yui NARUSE)
merge revision(s) 36544: [Backport #7017]

* win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

History

#1 - 07/25/2012 04:37 PM - nobu (Nobuyoshi Nakada)
Seems it occurs only in x64-mingw; and does not occur when loading the DLL by x64-mswin ruby.exe.
Needs more investigation.

#2 - 07/26/2012 05:35 AM - luislavena (Luis Lavena)
nobu (Nobuyoshi Nakada) wrote:

Seems it occurs only in x64-mingw; and does not occur when loading the DLL by x64-mswin ruby.exe.
Needs more investigation.

Do you think this simplification is good enough?

https://github.com/luislavena/experiments/tree/master/dll-handles

DllMain is only invoked once on this example, which might indicate something going on on Ruby.

After a GDB session, seems that dln.c line 1271 (LoadLibrary) is causing x64-msvcrts-ruby200.dll be loaded again.

Will keep investigating on this.

#3 - 07/27/2012 12:13 AM - phasis68 (Heesob Park)
After some investigating, I found this bug is due to exporting "DllMain" symbol of x64-msvcrts-ruby200.dll.

You can see the following line in the x64-msvcrts-ruby200.def.

VERSION 2.0
EXPORTS
DllMain
FD_CLR=rb_w32_fdclr
FD_ISSET=rb_w32_fdissset
...

I think that DllMain should not be exported.

Here is a simple workaround for win32/mkexports.rb

diff --git a/mkexports.rb b/mkexports.rb.new
index 3a1a91a..9dc8a19 100644
--- a/mkexports.rb
+++ b/mkexports.rb.new
@@ -38,6 +38,7 @@ class Exports
  winapis = {}
  syms_ = "ruby_sysinit_real" = "ruby_sysinit"
  each_export(objs) do |internal, export|
  next if internal == "DllMain" syms_ = export winapis[$1] = internal if $2 =~ internal end

With the patched and rebuilt version, I confirmed this bug is solved.

09/19/2021
Is this patch OK?

Index: win32/mkexports.rb
===================================================================
--- win32/mkexports.rb (revision 36541)
+++ win32/mkexports.rb (working copy)
@@ -116,7 +116,7 @@ class Exports::Mswin < Exports
is_data = $1
if noprefix or /@/ =~ l
next if /(?![
     @
]/ =~ l || /@
[\x{0061}-\x{007a}]{8,16}$/ =~ l ||/

/\(?::Init\|.*threadptr\|DllMain\)/ =~ l
/\(?::Init\|.*threadptr\|DllMain\)/ =~ l
l.sub!(/[@\_]/, '') if /@\d+/ =~ l elsif !l.sub!(/\+\d+(\@[\_]+)\((\@?`\'|\*\*)\$)/, ''

=end

#5 - 07/27/2012 12:15 PM - usa (Usaku NAKAMURA)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r36544.
Luis, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- win32/mkexports.rb: should not export DllMain(). reported by luis at [ruby-core:46743] [Bug #6790], solved by Heesob Park, and confirmed by nobu.

#6 - 07/27/2012 10:55 PM - luislavena (Luis Lavena)
Thank you Heesob Park, Usaku Nakamura and Nobu for fixing this!