

Ruby trunk - Bug #7681

Flip-flop test failure under MinGW

01/11/2013 04:42 AM - luislavena (Luis Lavena)

Status: Closed	
Priority: Normal	
Assignee: nobu (Nobuyoshi Nakada)	
Target version: 2.0.0	
ruby -v: ruby 2.0.0dev (2013-01-11 trunk 38770) [x64-mingw32]	Backport:
Description	
Hello,	
Since r38747 test_shared_thread is failing under both x86 and x64 MinGW (GCC 4.7.2):	
http://ci.rubyinstaller.org/view/All/job/ruby-trunk-x86-test-all/669/console	
http://ci.rubyinstaller.org/view/All/job/ruby-trunk-x64-test-all/545/console	
1) Failure: test_shared_thread(TestFlip) [C:/Users/Worker/Jenkins/workspace/ruby-trunk-x86-build/test/ruby/test_flip.rb:40]: flip-flop should be separated per threads. <[3, 4, 5]> expected but was <[3, 4]>.	
Related issues:	
Related to Ruby trunk - Bug #2618: Win32OLE RuntimeError due Colnitialize not...	Closed 01/20/2010

Associated revisions

Revision 0c61c3b6 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38848 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

Revision 38848 - 01/16/2013 09:45 AM - nobu (Nobuyoshi Nakada)

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]

History

#1 - 01/12/2013 07:56 AM - luislavena (Luis Lavena)

- Priority changed from 5 to 6

Ping?

Failed test might indicate something is not working as expected or test is doing something incorrectly.

This failed test is blocking automated builds that are provided to Windows users that want to try out Ruby 2.0 prior to the official release.

Not having automated builds will hit us hard as possible bugs are left uncovered in other platforms.

Please let me know if it is possible for you to review your change or someone else should be doing it.

Thank you.

#2 - 01/15/2013 11:33 AM - phasis68 (Heesob Park)

Ping?

I found this failure occurred after win32ole test.

Requiring win32ole affects this test.

The following code returns false.

```
def test_shared_thread
  require 'win32ole'
  ff = proc {|n| true if n==3..n==5}
  v = 1..9
  a = true
  th = Thread.new {
    v.select {|i|
      Thread.pass while a
      ff[i].tap {a = true}
    }
  }
  v1 = v.select {|i|
    Thread.pass until a
    ff[i].tap {a = false}
  }
  v2 = th.value
  v1==v2
end
```

This failure is raised from ext/win32ole/lib/win32ole.rb where redefines Thread#initialize like this:

```
class Thread
  alias :org_initialize :initialize
  def initialize(*arg, &block)
    if block
      org_initialize(*arg) {
        WIN32OLE.ole_initialize
        begin
          block.call(*arg)
        ensure
          WIN32OLE.ole_uninitialize
        end
      }
    else
      org_initialize(*arg)
    end
  end
end
```

end

#3 - 01/16/2013 10:28 AM - nobu (Nobuyoshi Nakada)

=begin

Seems a longstanding bug.

```
$ cat bug-7681.rb
class Bug7681 < Thread
  def initialize(*arg, &block)
    super(*arg) {yield(*arg)}
  end
end
```

```
$_ = '[Bug #7681]'
p Thread.new {$_}.value
p Bug7681.new {$_}.value
```

```
$ ruby-1.9.2 -v bug-7681.rb
ruby 1.9.2p323 (2012-05-21 revision 35743) [x86_64-darwin11]
nil
"[Bug #7681]"
```

```
$ /usr/bin/ruby -v bug-7681.rb
ruby 1.8.7 (2012-02-08 patchlevel 358) [universal-darwin11.0]
nil
"[Bug #7681]"
```

=end

#4 - 01/16/2013 06:39 PM - nobu (Nobuyoshi Nakada)

=begin

Now I'm uncertain if this is a bug.

I suspect it is same as the following code.

```
$ ruby -e 'class XThread;
  def initialize() @th = Thread.new{yield} end
  def value; @th.value; end;
end' -e '$="hoge" -e 'p XThread.new{$_}.value'
"hoge"
=end
```

#5 - 01/16/2013 06:45 PM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r38848](#).

Luis, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

win32ole.rb: use TracePoint

- ext/win32ole/lib/win32ole.rb: use TracePoint to hook all thread creation not only by Thread.new and to get rid of interference with svar scope. [Bug #7681] [ruby-core:51365]