Thread.stop and puts fail to work as documented.

07/23/2010 01:55 PM - phasis68 (Heesob Park)

Status: Closed
Priority: Normal
Assignee: kosaki (Motohiro KOSAKI)
Target version: 2.0.0
ruby -v: ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]

Backport:

Description
=begin
I have tested following three codes which are the sample code in RDoc Documentation.

t1.rb
====================================================
a = Thread.new { puts "a"; Thread.stop; puts "c" }
Thread.pass
puts "Got here"
a.run
a.join
sleep 1
====================================================

t2.rb
====================================================
a = Thread.new { print "a"; Thread.stop; print "c" }
Thread.pass
print "b"
a.run
a.join
sleep 1
====================================================

t3.rb
====================================================
c = Thread.new { Thread.stop; puts "hey!" }
c.wakeup
sleep 1
====================================================

Expected output

t1.rb
a
Got here
c

t2.rb
abc

hey!

on Ruby 1.8.6

$ ruby -v
ruby 1.8.6 (2009-06-08 patchlevel 369) [i686-linux]
$ ruby -v t1.rb
a
Got here
c
$ ruby -v t2.rb

abc$ ruby -v t3.rb
hey!
$

Result
t1.rb ==> success
t2.rb ==> success
t3.rb ==> success

on Ruby 1.9.1

$ ruby -v
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
$ ruby t1.rb
aGot here
c
$ ruby t2.rb
$ ruby t3.rb
$

Result
t1.rb ==> fail
t2.rb ==> success
t3.rb ==> fail

On Ruby 1.9.3dev
duometis02@Duo02:$ ruby -v
ruby 1.9.3.dev (2010-07-22 trunk 28707) [i686-linux]
$ ruby t1.rb
Got herea
t1.rb:5:in `join': deadlock detected (fatal)
  from t1.rb:5:in `<main>'
$ ruby t2.rb
bat2.rb:5:in `join': deadlock detected (fatal)
  from t2.rb:5:in `<main>'
$ ruby t3.rb
$

Result
t1.rb ==> fail
t2.rb ==> fail
t3.rb ==> fail

Interestingly, the following code sometimes works and sometimes failed on Ruby 1.9.3.dev.
t4.rb

====================================================
c = Thread.new { Thread.stop; puts "hey!" }
puts "Hi!"
c.wakeup
sleep 1
====================================================

$ ruby -v
ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]
$ ruby t4.rb
Hi!
$ ruby t4.rb
Hi!
hey!
$ ruby t4.rb
Hi!
hey!
$ ruby t4.rb
Hi!
$
Associated revisions

Revision b07b51f7 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32298 b2dd03c8-39d4-4d8f-98f8-823fe69b080e

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park
    phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32298 - 06/29/2011 03:47 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32299 b2dd03c8-39d4-4d8f-98f8-823fe69b080e

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)
  - thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is
    suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]
thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32299 - 06/29/2011 03:51 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 3a911701 - 06/29/2011 03:55 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32300 - 06/29/2011 03:55 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32300 - 06/29/2011 03:55 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32300 - 06/29/2011 03:55 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision 32300 - 06/29/2011 03:55 PM - kosaki (Motohiro KOSAKI)

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

Revision ed034f0c - 08/11/2011 12:38 AM - yugui (Yuki Sonoda)

merges r32298,r32299,r32300 and r32301 from trunk into ruby_1_9_2.

Rev 32298, r32299, r32300 and r32301 from trunk into ruby_1_9_2.

thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

thread.c (rb_thread_wakeup): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

thread.c (rb_thread_stop): change RDoc sample code. The old example is buggy and may not display anything by a race. The patch is suggested by Heesob Parrk phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]

thread.c (thread_s_pass): change RDoc description and remove a sample code. The actual implementation never behave as explained by an example. It's a documentation bug.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_2@32920 b2dd03c8-39d4-4d8f-98ff-823fe69b080e
On 23 Jul 2010, at 05:55, Heesob Park wrote:

on Ruby 1.8.6
$ ruby -v
ruby 1.8.6 (2009-06-08 patchlevel 369) [i686-linux]
$ ruby -v t1.rb
a
Got here
c
$ ruby -v t2.rb
abc$ ruby -v t3.rb
hey!
$

Result
t1.rb ==> success
t2.rb ==> success
t3.rb ==> success

This is because threads in 1.8.6 are not really threads. The engine is predictable within the realms of ruby frames.

on Ruby 1.9.1
$ ruby -v
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
$ ruby t1.rb
aGot here
c
$ ruby t2.rb
$ ruby t3.rb
$

Result
t1.rb ==> fail
t2.rb ==> success
t3.rb ==> fail

This could, or should vary, actually. The example code is not thread safe.

On Ruby 1.9.3dev
duometis02@Duo02:~$ ruby -v
ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]
$ ruby t1.rb
Got here

t1.rb:5:in `join': deadlock detected (fatal)
from t1.rb:5:in`'
$ ruby t2.rb
bat2.rb:5:in `join': deadlock detected (fatal)
from t2.rb:5:in`'
$ ruby t3.rb
$

So what's happening here is that you're getting to thread.join before the other thread has got to thread.stop. This is a race condition in your code. Threads are hard, and it's regarded that tools like stop, run, join, and pass are very very hard to use in a thread safe way. Assuming that calling Thread.pass with force at least a single expression to run in another thread is not valid.
Interestingly, the following code sometimes works and sometimes failed on Ruby 1.9.3.dev.

```ruby
t4.rb

```c = Thread.new { Thread.stop; puts "hey!" }
puts "Hi!"
c.wakeup
```

sleep 1

```$ ruby -v
ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]
$ ruby t4.rb
Hi!
$ ruby t4.rb
Hi!
$ ruby t4.rb
hey!
$ ruby t4.rb
Hi!
$ ruby t4.rb
Hey!
$ ruby t4.rb
Hi!
```

As above, this is a race condition. If `c.wakeup` is called before `Thread.stop`, then the code will never restart the thread after it stops itself. This is an error in the code, not in the interpreter.

=end

### #3 - 07/23/2010 10:25 PM - raggi (James Tucker)

On 23 Jul 2010, at 05:55, Heesob Park wrote:

Bug #3606: Thread.stop and puts fail to work as documented.
http://redmine.ruby-lang.org/issues/show/3606

Author: Heesob Park
Status: Open, Priority: Normal
Category: core, Target version: 1.9.x
ruby -v: ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]

I have tested following three codes which are the sample code in RDoc Documentation.

I see your point regarding this being code from the documentation. Examples as given in the documentation cannot work reliably with a preemptive scheduler and should be removed.

I would also say that use of `Thread.stop` and `Thread.pass` for any kind of "synchrony" should be very strongly discouraged, we have locking primitives.

=end

### #4 - 07/24/2010 09:53 AM - phasis68 (Heesob Park)

On 23 Jul 2010, at 05:55, Heesob Park wrote:

Bug #3606: Thread.stop and puts fail to work as documented.
http://redmine.ruby-lang.org/issues/show/3606

Author: Heesob Park
Status: Open, Priority: Normal
Category: core, Target version: 1.9.x
ruby -v: ruby 1.9.3dev (2010-07-22 trunk 28707) [i686-linux]

I have tested following three codes which are the sample code in RDoc Documentation.

I see your point regarding this being code from the documentation. Examples as given in the documentation cannot work reliably with a preemptive scheduler and should be removed.
I would also say that use of Thread.stop and Thread.pass for any kind of "synchrony" should be very strongly discouraged, we have locking primitives.

Thanks for your explanation.

I think that the sample code should be modified like this.

```
t1.rb
=====================================================================
a = Thread.new { puts "a"; Thread.stop; puts "c" }
sleep 0.1 while a.status!="sleep"
Thread.pass
puts "Got here"
a.run
a.join
sleep 1
=====================================================================

t2.rb
=====================================================================
a = Thread.new { print "a"; Thread.stop; print "c" }
sleep 0.1 while a.status!="sleep"
Thread.pass
print "b"
a.run
a.join
sleep 1
=====================================================================

t3.rb
=====================================================================
c = Thread.new { Thread.stop; puts "hey!" }
sleep 0.1 while c.status!="sleep"
c.wakeup
sleep 1
=====================================================================

Regards,
Park Heesob
=end

#5 - 07/24/2010 10:18 AM - raggi (James Tucker)
=begin

On 24 Jul 2010, at 01:52, Heesob Park wrote:

I think that the sample code should be modified like this.

```
t1.rb

a = Thread.new { puts "a"; Thread.stop; puts "c" }
sleep 0.1 while a.status!="sleep"
Thread.pass
puts "Got here"
a.run
a.join

sleep 1

Hmm status, yeah, I can't think of a better way there (I wish there was)

Maybe a.join(0.1) instead of sleep 0.1?

Thread.pass is not required anymore.

In docs for Thread.pass, I'm not sure what a good use case is.

=end
- Status changed from Open to Assigned
- Assignee set to kosaki (Motohiro KOSAKI)

#7 - 06/30/2011 12:47 AM - kosaki (Motohiro KOSAKI)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r32298.
Heesob, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- thread.c (rb_thread_run): change RDoc. The old example is buggy and may cause deadlock. The patch is suggested by Heesob Park phasis@gmail.com. Thank you! [Bug #3606][ruby-core:31454]