

Backport187 - Backport #2663

Hard hang (needs -9 to kill) in 1.8.7 build 248

01/27/2010 07:12 AM - wconrad (Wayne Conrad)

Status:	Closed
Priority:	Normal
Assignee:	
Description	
<p>=begin I've got a little piece of code that can hang Ruby 1.8.7 patchlevel 248 about one out of every 500 times it is run, on several (but not every) boxes. Here is foo.rb:</p>	
<pre>#!/usr/bin/ruby1.8 # The puts is not necessary to reproduce the problem. It just makes # it easy to tell when the problem happens. puts ARGV.first Thread.new do sleep 1 end system("#")</pre>	
<p>To reproduce the problem, execute foo.rb in a shell loop:</p>	
<pre>for i in `seq 10000` ; do ./foo.rb \$i ; done</pre>	
<p>The hang is hard: A TERM (15) signal won't stop it. A KILL (9) signal will.</p>	
<p>If it's going to hang, it will do it in 10,000 iterations. On the boxes I've tested, the hang is usually within the first 500 iterations.</p>	
<p>I've got five linux boxes that will show this behavior, and one that won't:</p>	
<ul style="list-style-type: none">• There are three fairly fast Intel 4-core boxes that can reproduce the problem. One is running Debian testing ("squeeze") and two running Debian unstable ("sid").• There is a moderate speed AMD two-core box that reproduces the problem, but takes longer. It runs Debian testing.• There is an old, slow single-core Intel box that does not show the problem. It runs Debian testing.	
<p>Although I found the problem using the Debian ruby/libruby packages, I confirmed that the problem happens when using a Ruby built without any of Debian's patches.</p>	
<p>Using the git tree at git://git.phusion.nl/ruby.git (branch v1_8_7_248) I used git bisect and found that the following commit introduced this problem:</p>	
<pre>commit d83cd902207920368dfe2de34a4be37fc774e6c8 Author: shyuhei shyouhei@b2dd03c8-39d4-4d8f-98ff-823fe69b080e Date: Tue Jul 14 11:31:37 2009 +0000</pre>	
<pre>merge revision(s) 23202,23268,23305: * eval.c (safe_mutex_lock): pthread_cleanup_push() must not be inside parens. * eval.c (rb_thread_start_timer): guard condition was inverted.</pre>	

```
[ruby-dev:38319]
* eval.c (get_ts): use readtime clock. [ruby-dev:38354]
* eval.c (rb_thread_stop_timer): clear thread_init while
```

locking.

This is a small commit that doesn't change very many lines, so I was able to eliminate all but a single changed line as the cause of this problem:

```
--- a/eval.c
+++ b/eval.c
@@ -12316,7 +12316,7 @@ rb_thread_start_timer()
void *args[2];
static pthread_cond_t start = PTHREAD_COND_INITIALIZER;
```

- if (!thread_init) return;
- if (thread_init) return; args[0] = &time_thread; args[1] = &start; safe_mutex_lock(&time_thread.lock);

I think this is CVS rev 23268, ruby-dev:38319, bug [#1402](#).

I confirmed that this line has something to do with the problem by reverse-applying it to build 248 and seeing that the problem no longer occurs. But rev 23268 seems correct to my untrained eye, so perhaps the real problem is somewhere else.

Best Regards,
Wayne Conrad
=end

Related issues:

Related to Ruby master - Bug #270: lazy timer thraed creation	Closed	
Is duplicate of Backport187 - Backport #2603: NetBSD 5.0 pthread	Closed	01/14/2010
Is duplicate of Ruby 1.8 - Bug #1872: [ruby_1_8] Kernel#system doesn't work i...	Closed	08/04/2009

History

#1 - 01/27/2010 04:10 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

```
=begin
r23268 is correct fix, but it realize the bug around Ruby 1.8's thread.
Fix for it is applied in ruby_1_8 by #2603 and those fix will backport to Ruby 1.8.7.
Please try ruby_1_8 branch.
=end
```

#2 - 06/08/2010 01:07 PM - shyouhei (Shyouhei Urabe)

- Status changed from Assigned to Closed

```
=begin
This issue was solved with changeset r28203.
Takahiro, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.
=end
```