Ruby - Bug #7648

GServer does not close cleanly from signal interrupt context

01/03/2013 10:35 AM - jleo3 (Joe Leo)

Status:	Rejected		
Priority:	Normal		
Assignee:	kosaki (Motohiro KOSAKI)		
Target version:	2.6		
ruby -v:	ruby 2.0.0dev (2013-01-02 trunk 38676) [i686-linux]	**Backport:**	

Description

SUMMARY:
When a signal interrupt is trapped, we can no longer call #close on GServer without it throwing a ThreadError.

STEPS TO REPEAT:

1. Run the following code:

```
require 'gserver'

server = GServer.new 8080
server.start
trap("SIGINT") { server.stop }
gets # or any command that keeps the process running
```

2. Hit CTRL+C or whichever command will send the interrupt signal to this program.

WHAT I EXPECTED: In version 1.9.3, CTRL+C sends an interrupt signal and the program exits cleanly.

WHAT HAPPENED: When running the version from trunk the following stack trace is thrown.

^C/home/joe/.rvm/rubies/ruby-head/lib/ruby/2.0.0/gserver.rb:116:in synchronize': can't be called from trap context (ThreadError) from /home/joe/.rvm/rubies/ruby-head/lib/ruby/2.0.0/gserver.rb:116:in stop'
from gserver_bug.rb:5:in block in <main>' from gserver_bug.rb:6:in call'
from gserver_bug.rb:6:in gets' from gserver_bug.rb:6:in gets'
from gserver_bug.rb:6:in `'

POSSIBLY RELEVANT: https://bugs.ruby-lang.org/issues/6416

NOTE: This was tried with AND without RVM with the same results.

History

#1 - 01/14/2013 01:00 AM - jleo3 (Joe Leo)

Confirming that this is still an issue on RC1:

ruby -v
ruby 2.0.0dev (2013-01-07 trunk 38733) [i686-linux]

/home/joe/lib/lib/ruby/2.0.0/gserver.rb:116:in synchronize': can't be called from trap context (ThreadError) from /home/joe/lib/lib/ruby/2.0.0/gserver.rb:116:in stop'
from /home/joe/dev/bane/lib/bane/launcher.rb:19:in block in stop' from /home/joe/dev/bane/lib/bane/launcher.rb:19:in each'
from /home/joe/dev/bane/lib/bane/launcher.rb:19:in stop' from ./bin/bane:22:in block in '
from /home/joe/lib/lib/ruby/2.0.0/gserver.rb:140:in call' from /home/joe/lib/lib/ruby/2.0.0/gserver.rb:140:in join'
from /home/joe/lib/lib/ruby/2.0.0/gserver.rb:140:in join' from /home/joe/dev/bane/lib/bane/launcher.rb:15:in block in join'
from /home/joe/dev/bane/lib/bane/launcher.rb:15:in each' from /home/joe/dev/bane/lib/bane/launcher.rb:15:in join'
from ./bin/bane:23:in `'

#2 - 01/25/2013 12:16 PM - ko1 (Koichi Sasada)

- Category set to lib

- Target version set to 2.0.0

Who can check it?

#3 - 02/17/2013 01:51 PM - ko1 (Koichi Sasada)

- Assignee set to mame (Yusuke Endoh)

No response here.

#4 - 02/17/2013 02:45 PM - mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 2.6

#5 - 05/31/2013 01:17 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee changed from mame (Yusuke Endoh) to kosaki (Motohiro KOSAKI)

kosaki-san, what do you think?

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 08/31/2013 02:20 AM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Rejected

Holding mutex in trap is deadlockable. It is what ruby complained. The best workaround is to make new thread in trap handler and stop gserver asynchnorously, I think.