

Ruby trunk - Bug #7917

Can't write to a Logger in a signal handler

02/23/2013 09:49 AM - mperham (Mike Perham)

Status:	Rejected	
Priority:	Normal	
Assignee:	kosaki (Motohiro KOSAKI)	
Target version:	2.1.0	
ruby -v:	2.0.0-rc2	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN
Description		
Looks like Ruby 2.0 does not allow Mutex#lock within a signal handler. This prevents Logger from working since it uses an underlying mutex.		
log writing failed. can't be called from trap context log writing failed. can't be called from trap context log writing failed. can't be called from trap context log writing failed. can't be called from trap context		
Here's Ruby code which reproduces the problem:		
require 'logger'		
LOG = Logger.new(STDOUT)		
LOG.info "Now logging!"		
trap 'INT' do		
LOG.info "Hello"		
end		
sleep		
LOG.info "Done"		
Run it and hit Ctrl-C.		

History

#1 - 02/23/2013 10:40 AM - kosaki (Motohiro KOSAKI)

Because your example is deadlockable when using 1.9.3 or earlier. 2.0 detect your mistake and tell you.

#2 - 02/23/2013 03:22 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to kosaki (Motohiro KOSAKI)

- Target version set to 2.1.0

This issue is arguable, but anyway I don't consider it a showstopper.
I postpone it to 2.1.0.

It is very arguable if the fault is in the code, or in the design of Ruby core.

--

Yusuke Endoh mame@tsg.ne.jp

#3 - 02/23/2013 05:23 PM - normalperson (Eric Wong)

"mperham (Mike Perham)" mperham@gmail.com wrote:

```
trap 'INT' do
  LOG.info "Hello"
end
```

An ugly workaround I use is to spawn a thread inside trap:

```
trap 'INT' do
Thread.new do
LOG.info "Hello"
end
end
```

#4 - 02/23/2013 05:23 PM - kosaki (Motohiro KOSAKI)

On Sat, Feb 23, 2013 at 3:06 AM, Eric Wong normalperson@yhbt.net wrote:

"mperham (Mike Perham)" mperham@gmail.com wrote:

```
trap 'INT' do
LOG.info "Hello"
end
```

An ugly workaround I use is to spawn a thread inside trap:

```
trap 'INT' do
Thread.new do
LOG.info "Hello"
end
end
```

Um. this is safe.

btw, following is unsafe when 1.9.x. because trap safe th.join is a new feature of 2.0.

```
trap 'INT' do
t = Thread.new do
LOG.info "Hello"
end
t.join
end
```

see <http://bugs.ruby-lang.org/issues/6416>

Hm.
Should we backport this fix to 1.9.3 too?

#5 - 02/23/2013 05:23 PM - kosaki (Motohiro KOSAKI)

```
trap 'INT' do
t = Thread.new do
LOG.info "Hello"
end
t.join
end
```

Oops, no. This code is completely wrong. Sorry for confusing.
Please don't mind my previous mail. This code is deadlockable and should never be used.

#6 - 03/13/2013 04:38 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Rejected

Closed. because it's not a bug.

#7 - 12/22/2017 01:05 PM - hkmaly (Honza Maly)

What IS the recommended way to fix this if Mutex nor Thread are safe in signal handler? Also created question for what is safe in signal handler in general as [#14222](#)

#8 - 12/23/2017 01:12 AM - normalperson (Eric Wong)

hkmaly@matfyz.cz wrote:

What IS the recommended way to fix this if Mutex nor Thread

are safe in signal handler? Also created question for what is safe in signal handler in general as [#14222](#)

See my response for [ruby-core:84410] [Misc [#14222](#)] :->
<https://bugs.ruby-lang.org/issues/14222>