Ruby master - Bug #7212

"stack level too deep" in Fiber much earlier in new versions of 1.9.3

10/24/2012 11:21 PM - fiddur (Fredrik Liljegren)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:	2.0.0		
ruby -v:	ruby 1.9.3p286 (2012-10-12 revision 37165) [x86_64-linux] (and others, see description)	**Backport:**	

Description

I was getting SystemStackError in my application on some servers and not others; on all with 1.9.3-p286 and on some with p194, and on none with 1.9.2-head. I boiled it down to this:

```
def recursive(level = 0)
-> do
p "In block #{level}"
if level < 1000
subblock = recursive(level + 1)
subblock.call
end
end
end

p "Doing recursive call in a fiber"
Fiber.new { recursive.call }.resume
```

On server A, 1.9.3-p194 and 1.9.3-p286 got up to 11 levels of recursion, while 1.9.2-head got up to 97 levels.
On server B (without 1.9.2), 1.9.3-p194 got 55 levels while p286 still got 11 levels.

I don't know what changes are made, but I think 11 levels are way on the low side for many applications.

(My original problem was with a thin-server running rack-fiber_pool with em-synchrony getting too deep in a regexp in Addressable::URI.)

History

#1 - 10/25/2012 10:21 AM - usa (Usaku NAKAMURA)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

#2 - 10/25/2012 01:56 PM - fiddur (Fredrik Liljegren)

Here's a little more debug-info (not sure if it's needed, but anyhow...)

I tried the patch from #3187, increasing the stack size of fibers. I confirmed that the patch is working on a normal recursing method (without lambda block), increasing recursability 4 times by setting 16kb stacksize. That did NOT affect this bug at all, still on 11 levels of recursion.

Not knowing how it's implemented, I tried lambda-blocks without closure-behaviour; i.e. without using external variables, using just:

```
def recursive
-> do
p "Block"
subblock = recursive
subblock.call
end
end
```

...but it's exactly the same.

I also tried without the lambda notation, just sending in a block to recursive method. That got me up to level 250 on both 1.9.3-p194 and 1.9.2-p320. That's using:

```
def recursive(level=0, &block)
p "Level #{level}"
```

```
recursive(level+1) { block.call }
end
Fiber.new { recursive {} }.resume
```

Well, hope it helps.

#3 - 10/30/2012 09:17 AM - ko1 (Koichi Sasada)

- Target version set to 2.0.0

#4 - 12/20/2012 07:38 AM - ko1 (Koichi Sasada)

- Status changed from Assigned to Feedback

Do you use same compiler and compile option on each environments?

#5 - 02/13/2013 03:57 PM - ko1 (Koichi Sasada)

- Status changed from Feedback to Closed

No feedback.

#6 - 03/06/2013 06:31 PM - fiddur (Fredrik Liljegren)

ko1 (Koichi Sasada) wrote:

> Do you use same compiler and compile option on each environments?

Yes, it was compiled with rvm with no options specified.

I'm sorry for the late answer, for some reason I didn't get any mail notification even though my settings are to get email for things "I watch or I'm am involved in".

Files

recursive.rb	228 Bytes	10/24/2012	fiddur (Fredrik Liljegren)