

Ruby master - Bug #9664

cannot resume transferred Fiber even if it should resume

03/23/2014 06:06 AM - royaltm (Rafał Michalski)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: 2.1.0 2.0.0-p353	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN
Description	
The simplified case:	
<pre>root = Fiber.current f = Fiber.new { puts 'transfer' root.transfer puts 'yield' Fiber.yield puts 'ok' } f.resume f.transfer f.resume</pre>	
First we resume into a new fiber, then the fiber f transfers control back to root and then root transfers control back to f. Then the fiber f yields back to root and when root tries to resume fiber f the unexpected error is raised.	
In ruby MRI 1.9.(1-3) it works as expected and the result is:	
<pre>transfer yield ok</pre>	
However since MRI 2.0 we get:	
<pre>transfer yield -:11:in `resume': cannot resume transferred Fiber (FiberError) from -:11:in `<main>'</pre>	
In ruby #transfer docs one reads:	
The fiber which receives the transfer call is treats it much like a resume call. Arguments passed to transfer are treated like those passed to resume.	
You cannot resume a fiber that transferred control to another one. This will cause a double resume error. You need to transfer control back to this fiber before it can yield and resume.	
But it looks like since 2.0 you can't yield and resume fiber after transferring control from or back to it. This looks like if a fiber has ever transfered control with #transfer then it is somehow marked and resuming to it is no more possible.	
This bug kills possibility to use e.g. fiber-synchronized async frameworks together with enumerators.	

History

#1 - 03/24/2014 07:08 PM - royaltm (Rafał Michalski)

More straightforward example and a convenient way to detect the bug:

```
begin
  Fiber.new { Fiber.yield Fiber.current }.transfer.resume
  puts "Cool I'm in ruby 1.9.x. I've got fibers!"
end
```

```
rescue FiberError
  puts "Too bad, I'm in 2.x and no transfer/yield/resume for me. Is it supposed to be a feature? :("
end
```

#2 - 12/01/2017 09:40 PM - cremes (Chuck Remes)

This bug or limitation still exists on Ruby 2.4.2. Any plans to address it?

This thread on [stackoverflow.com](https://stackoverflow.com/questions/38199283/ruby-fiber-resuming-transferred-fibers) identifies a possible bug in the C code.

<https://stackoverflow.com/questions/38199283/ruby-fiber-resuming-transferred-fibers>

```
fib->transferred = 1;
```

The transferred flag is set to 1 but it is never cleared under any circumstance.