# Bug #16509

gem - Bug #16509

puts from a thread seem to be missing linefeed when run in ruby 2.7.0 console

01/14/2020 09:33 PM - jmaino (Jim Maino)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>2.7.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>2.7.0</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED</td>
</tr>
</tbody>
</table>

## Description

This is strange, and I'm hoping I explain it sufficiently - but I've upgraded to ruby 2.7.0 and I'm seeing missing linefeeds in puts outputs from EM that's within a thread.

I did not see this problem when running in ruby 2.6.5's console output - it seems to be related to my move to 2.7.0.

I've simplified this to reproduce this as follows. If I run the following EM without being inside of a thread, then I get an OK output - example:

```ruby
irb(main):042:1* EventMachine.run do
  EM.add_periodic_timer(1) { puts "Tick ..." }
  EM.add_timer(3) do
    puts "I waited 3 seconds"
    EM.stop_event_loop
  end
irb(main):049:0> end
```

Tick ...
Tick ...
I waited 3 seconds
=> nil

Note all of the strings that are output from the puts commands within the EM loop are fully left-justified in the output. But if I wrap this in a thread, the output lines appear to be missing the linefeed, but do have the CR - such that the successive puts strings are not left justified as they sequence, and build from the X-offset that ended the previous puts string output.

```ruby
irb(main):001:1* Thread.new do
  EM.add_periodic_timer(1) { puts "Tick ..." }
  EM.add_timer(3) do
    puts "I waited 3 seconds"
    EM.stop_event_loop
  end
irb(main):009:0> end
```

_______Tick ...
I waited 3 seconds
=> #

Is there something that's known to have changed that is causing this? Can others reproduce this as well?

I can't tell is this is EM's or ruby's - so any help would really be appreciated. However, the folks over at eventmachine think this belongs to ruby and not themselves.

I only see this from puts statements within the EM loop within a Thread. If the wrapping Thread is not there, there's no issue. If there's just a sequence of puts in a Thread and no EM loop, there's no issue.

It seems as if there's something about the combo of a Thread wrapping an EM loop that's making the underlying puts misbehave.

02/23/2020
Thanks,
-Jim

Associated revisions

Revision 4e56ec4e - 01/17/2020 03:15 PM - nobu (Nobuyoshi Nakada)
[ruby/io-console] Set OPOST when intr is true

To enable implementation-defined output processing, for the compatibility with readline. [Bug #16509]

https://bugs.ruby-lang.org/issues/16509
https://github.com/ruby/io-console/commit/8c8b0b6757

History

#1 - 01/16/2020 06:27 PM - Marthyn (Marthyn Olthof)

Ruby 2.7.0

```ruby
irb(main):002:1* EventMachine.run do
  irb(main):003:1*   EM.add_periodic_timer(1) { puts "Tick ..." }
  irb(main):004:2*   EM.add_timer(3) do
  irb(main):005:2*     puts "I waited 3 seconds"
  irb(main):006:2*     EM.stop_event_loop
  irb(main):007:1*     end
  irb(main):008:0> end
Tick ...
I waited 3 seconds
=> nil
irb(main):009:0> Thread.new do
  irb(main):010:1*   EventMachine.run do
  irb(main):011:2*     EM.add_periodic_timer(1) { puts "Tick ..." }
  irb(main):013:3*     EM.add_timer(3) do
  irb(main):014:3*       puts "I waited 3 seconds"
  irb(main):015:3*       EM.stop_event_loop
  irb(main):016:2*     end
  irb(main):017:1*     end
  irb(main):018:0> end
=> #<Thread:0x00007f8e4ea08c08 (irb):9 run>
irb(main):019:0> Tick ...
I waited 3 seconds
```

Same result

Ruby 2.6.5

```ruby
irb(main):002:0> EventMachine.run do
  irb(main):003:1*   EM.add_periodic_timer(1) { puts "Tick ..." }
  irb(main):004:1>   EM.add_timer(3) do
  irb(main):005:2*     puts "I waited 3 seconds"
  irb(main):006:2>     EM.stop_event_loop
  irb(main):007:2>   end
  irb(main):008:1> end
Tick ...
I waited 3 seconds
=> nil
irb(main):009:0> Thread.new do
  irb(main):010:0>   EventMachine.run do
  irb(main):011:1*     EM.add_periodic_timer(1) { puts "Tick ..." }
  irb(main):013:2*     EM.add_timer(3) do
  irb(main):014:3*       puts "I waited 3 seconds"
  irb(main):015:3>      EM.stop_event_loop
  irb(main):016:3>    end
  irb(main):017:2>    end
  irb(main):018:1> end
=> #<Thread:0x00007fa03f108d80 @(irb):10 run>
irb(main):019:0> Tick ...
```
Tick ...
I waited 3 seconds
Tick ...

No indent.

#2 - 01/17/2020 03:17 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset git|4e56ec4e7744d4cb407ca8e095e0d0d12fa.

[ruby/io-console] Set OPOST when intr is true

To enable implementation-defined output processing, for the compatibility with readline. [Bug #16509]

https://bugs.ruby-lang.org/issues/16509
https://github.com/ruby/io-console/commit/8c8b0b6757

#3 - 01/17/2020 03:20 PM - nobu (Nobuyoshi Nakada)
- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED
- Subject changed from puts from EM within a thread seem to be missing linefeed when run in ruby 2.7.0 console to puts from a thread seem to be missing linefeed when run in ruby 2.7.0 console