

Ruby - Bug #14542

syswrite spuriously empties the given string

02/22/2018 01:56 PM - sylvain.joyeux (Sylvain Joyeux)

Status:	Third Party's Issue	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	2.3.6, 2.4.3	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description <p>I'm doing a bit of buffer management to write on a socket. The main method looks like this:</p> <pre>@write_buffer.concat(new_bytes) if new_bytes written_bytes = io.syswrite(@write_buffer) @write_buffer = @write_buffer[written_bytes..-1] !@write_buffer.empty?</pre> <p>However, in some cases, I get a nil @write_buffer. I've added some debugging info:</p> <pre>@write_buffer.concat(new_bytes) if new_bytes size_before_syswrite = @write_buffer.size written_bytes = io.syswrite(@write_buffer) size_after_syswrite = @write_buffer.size updated_write_buffer = @write_buffer[written_bytes..-1] if !updated_write_buffer STDERR.puts "NIL WRITE BUFFER, size_before_syswrite=#{size_before_syswrite} size_after_syswrite= #{size_after_syswrite} written_bytes=#{written_bytes}" end @write_buffer = updated_write_buffer !@write_buffer.empty?</pre> <p>in all cases, I got the following output:</p> <pre>NIL WRITE BUFFER, size_before_syswrite=3106 size_after_syswrite=0 written_bytes=3106</pre> <p>However, I could not reproduce the problem with toy setups. syswrite's string handling is beyond the little I know of MRI's internals ;-)</p>		

History

#1 - 02/22/2018 11:03 PM - normalperson (Eric Wong)

sylvain.joyeux@m4x.org wrote:

<https://bugs.ruby-lang.org/issues/14542>

- Target version:
- ruby -v: 2.3.6, 2.4.3

Hitting this in 2.4.3 might be my fault, but I'm not sure about Ruby 2.3.6 since my garbage reduction fixes aren't in that.

For 2.4.3, can you try removing the `rb_str_tmp_frozen_release` call in the `rb_io_syswrite` function in `io.c`?

I'm doing a bit of buffer management to write on a socket. The main method looks like this:

```
@write_buffer.concat(new_bytes) if new_bytes
written_bytes = io.syswrite(@write_buffer)
@write_buffer = @write_buffer[written_bytes..-1]
!@write_buffer.empty?
```

Just to cover all bases, are you somehow sharing `@write_buffer`

between multiple threads? Or do you have any signal handler where `@write_buffer` may be clobbered?

I'll try to be around this weekend to do some deeper investigation.

#2 - 02/23/2018 11:41 AM - sylvain.joyeux (Sylvain Joyeux)

For 2.4.3, can you try removing the `rb_str_tmp_frozen_release` call in the `rb_io_syswrite` function in `io.c`?

With difficulty ... the only place I manage to reproduce semi-reliably is on Travis. This does happen outside Travis, but rarely. It happens that the test suite triggers it on Travis pretty often for whatever reason.

Just to cover all bases, are you somehow sharing `@write_buffer` between multiple threads?

It shouldn't, but there *are* multiple threads in the system. I'll check it out just to be sure.

Or do you have any signal handler where `@write_buffer` may be clobbered?

No signal handler.

#3 - 02/26/2018 12:32 PM - sylvain.joyeux (Sylvain Joyeux)

It seems so far that it is indeed a multithreading problem on my side, possibly specific to the test suite. My apologies for the noise, and thanks a lot to Eric for his help.

I don't find a way to close this ...

#4 - 12/19/2018 11:33 PM - sylvain.joyeux (Sylvain Joyeux)

Could someone close this report ? It's been solved as 'not a bug in Ruby'

#5 - 12/20/2018 12:16 AM - Eregon (Benoit Daloze)

- *Status changed from Open to Third Party's Issue*