

Ruby - Bug #21037

Ractors hang with multiple threads

01/14/2025 09:08 PM - tenderlovmaking (Aaron Patterson)

Status:	Closed	
Priority:	Normal	
Assignee:	ractor	
Target version:		
ruby -v:	ruby 3.5.0dev (2025-01-14T20:59:39Z master f1e32914eb) +PRISM [x86_64-linux]	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

The following program hangs, but I don't expect it to hang:

```
# frozen_string_literal: true

class Map
  def initialize
    @r = Ractor.new {
      cache = { "int" => "integer", "bool" => "boolean" }
      loop do
        key = receive
        Ractor.yield key
      end
    }
    freeze
  end

  def fetch(key)
    @r.send key
    m = @r.take
    raise unless key == m
    m
  end
end

tm = Map.new
t1 = Thread.new { 10.times { |i| p [i + 1, tm.fetch("int")] } }
t2 = Thread.new { 10.times { |i| p [i + 1, tm.fetch("bool")] } }

t1.join
t2.join
```

I'm not sure why this program hangs yet, but it does both on macOS and Linux. I've tested with master, Ruby 3.4, and Ruby 3.3. I'm not sure why it's hanging yet, but I wanted to file an issue.

Related issues:

Related to Ruby - Bug #17624: Ractor.receive is not thread-safe Closed

History

#1 - 01/15/2025 07:30 PM - luke-gru (Luke Gruber)

Maybe this fixes it? <https://github.com/ruby/ruby/pull/12520>

Edit: I just tried with this branch but it seems unrelated. If you run with RUBY_DEBUG it fails an assertion having to do with ractor sleep (the receive call). I'll take a look when I get a chance.

#2 - 01/15/2025 08:08 PM - luke-gru (Luke Gruber)

It seems calling ractor.take from multiple threads is unsupported at the moment. I don't know if this is a limitation or a bug. Of course, you can work around it and call select or take from a single thread, like main, and signal a ready event with a queue or something for the consumer threads.

Edit: There are some TODOs in the codebase having to do with waiting threads for ractors, and they explicitly say "TODO: make multithreaded". I believe this is a known problem, at least for [@ko1 \(Koichi Sasada\)](#).

#3 - 01/24/2025 09:52 PM - luke-gru (Luke Gruber)

I created a PR here: <https://github.com/ruby/ruby/pull/12633>. I don't know how to get it working for Win32 builds yet, I'll have to take a closer look at that.

Edit: Everything works now except for Visual Studio 2022, I'm trying to setup a Windows VM to see what's failing.

#4 - 05/08/2025 10:38 PM - jhawthorn (John Hawthorn)

- Assignee set to ractor

#5 - 05/12/2025 11:16 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#6 - 05/13/2025 07:17 PM - byroot (Jean Boussier)

- Related to Bug #17624: Ractor.receive is not thread-safe added

#7 - 05/14/2025 05:02 PM - luke-gru (Luke Gruber)

I didn't tag the commit message properly, but this can be closed. PR <https://github.com/ruby/ruby/pull/12633> was merged. Thanks!

#8 - 05/14/2025 05:31 PM - tenderlovmaking (Aaron Patterson)

- Status changed from Assigned to Closed

Fixed in 1d4822a