

Ruby - Bug #21108

C-c (SIGINT) crashes ruby when looping Ractors are not taken?

02/02/2025 08:59 AM - osyoyu (Daisuke Aritomo)

Status:	Closed	
Priority:	Normal	
Assignee:	ractor	
Target version:		
ruby -v:	ruby 3.5.0dev (2025-01-27T08:19:32Z master c3c7300b89) +PRISM [x86_64-linux]	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

Hitting C-c causes ruby to crash under these circumstances.

Repro

Hit C-c while the program is running.

```
r1 = Ractor.new do
  ary = []
  loop do
    Ractor.receive
    Ractor.yield ary.push(1)
  end
end

r2 = Ractor.new(r1) do |r1|
  loop do |i|
    r1.send(nil)
    r1.take
  end
end

# r1.take # `take`ing r1 will fix this issue
r2.take
```

Expected

The process quits.

Actual

An [BUG] unreachable occurs.

The rb_bug() is called inside ractor_try_yield.
<https://github.com/ruby/ruby/blob/ace39a3ed40cc32ff8d48893173413a7e05b8316/ractor.c#L1359>

```
wrap.rb:1: warning: Ractor is experimental, and the behavior may change in future versions of Ruby
! Also there are many implementation issues.
<internal:ractor>:644: [BUG] unreachable
ruby 3.5.0dev (2025-01-27T08:19:32Z master c3c7300b89) +PRISM [x86_64-linux]

-- Control frame information -----
c:0005 p:0003 s:0022 e:000021 METHOD <internal:ractor>:644
c:0004 p:0014 s:0015 e:000014 BLOCK wrap.rb:5
c:0003 p:0017 s:0012 e:000011 METHOD <internal:kernel>:168
c:0002 p:0008 s:0007 e:000006 BLOCK wrap.rb:3 [FINISH]
c:0001 p:---- s:0003 e:000002 DUMMY [FINISH]

-- Ruby level backtrace information -----
wrap.rb:3:in 'block in <main>'
```

```
<internal:kernel>:168:in 'loop'  
wrap.rb:5:in 'block (2 levels) in <main>'  
<internal:ractor>:644:in 'yield'
```

```
-- Threading information -----  
Total ractor count: 2  
Ruby thread count for this ractor: 1
```

```
-- C level backtrace information -----  
ruby(0x639a516dc55a) [0x639a516dc55a]  
[0x639a512af8dd]  
[0x639a512a3a31]  
[0x639a513876a1]  
[0x639a513877a6]  
[0x639a5149b0b3]  
[0x639a51487565]  
[0x639a514317db]  
[0x639a51433e71]  
[0x639a51434295]
```

See full log at: <https://gist.github.com/osyoyu/2ae4fe3e41c139703bab5c321a81dda5>

History

#1 - 02/02/2025 10:22 AM - osyoyu (Daisuke Aritomo)

Created a patch: <https://github.com/ruby/ruby/pull/12689>

#2 - 02/02/2025 11:19 PM - luke-gru (Luke Gruber)

This is actually related to <https://bugs.ruby-lang.org/issues/21090>, my patch fixes this issue as well.

#3 - 02/03/2025 03:47 AM - osyoyu (Daisuke Aritomo)

Thank you for your comment! Your patch looks promising. Hope it gets merged soon.

#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 - 06/05/2025 11:39 PM - jhawthorn (John Hawthorn)

- Status changed from Assigned to Closed

I believe this should be fixed as Ractor.yield no longer exists under the Ractor::Port API