Ruby - Bug #17587

Segmentation fault with ractors and unix signals

01/27/2021 08:56 PM - mweitzel (Matthew Weitzel)

Status:	Rejected		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:			
ruby -v:	ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]	**Backport:**	2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN

Description

Segmentation fault when trapping signals and using a Ractor.

Can be reproduced by running the following

```
Ractor.new do
  Signal.trap('INT') do
    Ractor.yield("yoo hoO! big summer blowout")
  end
  `kill -2 #{$$}`
  `kill -2 #{$$}`
  `kill -2 #{$$}`
end.take
```

History

#1 - 01/29/2021 09:39 AM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

current master is stuck.
I'll check it.

#2 - 02/07/2021 10:03 AM - wanabe (_ wanabe)

The current behavior of stacking appears to be as expected.

The registered signal handler is called from the main thread.
https://git.ruby-lang.org/ruby.git/tree/thread.c?id=947d93b715436b13eefa39f87737bdad3c1f870a#n2430
Ractor.yield in the main thread must cause stuck because there are no Ractor to take it in parallel.
In addition, signals are basically ignored in the signal handler, because of interrupt_mask. https://bugs.ruby-lang.org/issues/6009
For example, Signal.trap("USR2") do sleep end; Process.kill(:USR2, $$) can make ruby stuck.

I guess this is normal behavior that has nothing to do with Ractor.
(However, there is still room for consideration of safe behavior when a signal is received in a signal handler.)

#3 - 11/30/2021 05:26 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#4 - 12/14/2021 04:50 PM - ko1 (Koichi Sasada)

- Status changed from Assigned to Rejected

as @wanabe (_ wanabe) san described, it is an expected behavior.
(I'm not sure why I prohibited trap in Ractor...)

Files

ruby_2021-01-27-135454_Matthews-MacBook-Air.crash	29.6 KB	01/27/2021	mweitzel (Matthew Weitzel)