Ruby - Bug #21146

VM_ASSERT(expr) gives bad bug report results when another ractor fails an assertion during printing of report

02/17/2025 11:19 PM - luke-gru (Luke Gruber)

Status:	Assigned		
Priority:	Normal		
Assignee:	ractor		
Target version:			
ruby -v:		**Backport:**	3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

test.rb:

```
rs = 100.times.map do
  Ractor.new do
    cnt = rand 3
    cnt += 1 if cnt.zero?
    sleep cnt
    100.times do |i|
      if i != 0 && i % 50 == 0
        Ractor.fail_assert
      end
    end
  end
end
```

ractor.rb:

```
def self.fail_assert
  __builtin_cexpr! %q{
    VM_ASSERT(0), Qfalse
  }
end
```

make run

I would like to be able to see the bug report for the first failed assertion, without any output from the other ractors.

History

#1 - 02/18/2025 12:13 AM - luke-gru (Luke Gruber)

PR here: https://github.com/ruby/ruby/pull/12770

#2 - 03/11/2025 07:56 AM - ko1 (Koichi Sasada)

Your patch uses RB_VM_LOCK_ENTER_NO_BARRIER but it should block normal use of rb_bug() (using rb_bug() is irregular case though).
So I think it should use simpler mechanism to synchronize rb_bug() calling. For example, introducing a global variable to avoid multiple rb_bug() calls.

(btw VM_ASSERT() calls rb_bug() if RUBY_DEBUG (or other macros) is defined, so rb_bug() is suitable for the example)

#3 - 03/11/2025 09:47 PM - luke-gru (Luke Gruber)

Thanks for your comment. I can make it simpler, but I am a bit confused as to what I should do instead. If the first thread gets to the global variable first and enters rb_vm_bugreport, my thinking was that other threads that also try to enter this function should be blocked (mutex, sleep, etc.). Are you saying just return from the function and let the other thread continue anyway?

Also when you say use a global variable, do you mean an atomic global? I'm open to doing whatever you want, because maybe I'm overthinking it for just a debug case anyway.

Thanks again!

#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