**Ruby master - Bug #18903**

**Stack overflow signal handling seems to be triggered once and then not working after**

07/07/2022 02:33 PM - chrisseaton (Chris Seaton)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [arm64-darwin21]</td>
</tr>
</tbody>
</table>

**Backport:** 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

**Description**

This program creates a recursive object graph and then tries to convert it to JSON with no max depth, so it stack overflows in C code that does no co-operative stack overflow checks, as the bytecode interpreter would. This therefore triggers a segmentation fault and the stack overflow detection there. It works the first time, but the second time around it doesn't and the program hard crashes on M1.

Is there something like a guard page permission that is switched during the handling, and needs to switched back for the guard page to work again?

Note that it isn't JSON specific - I think any stack overflow within C code would do it.

```ruby
require 'json'
a = []
a << a

begin
  JSON.dump(a)
rescue Exception
  puts 'rescued'
end

JSON.dump(a)
```

**History**

**#1 - 07/07/2022 02:33 PM - chrisseaton (Chris Seaton)**

(Found by Jean Boussier)

**#2 - 07/07/2022 08:28 PM - Eregon (Benoit Daloze)**

In general it is not possible to recover from a stack overflow (be it in C or Ruby), the executing program should be considered hopelessly corrupted because e.g. it might have happened in the middle of a critical section.

So IMHO the right fix would be to make stack overflows not rescue-able and hard exit on such a case.

That's from my experience with stack overflows on the JVM, some details are different on CRuby but the overall problem is the same.