**Ruby master - Bug #13163**

Uncaught exceptions may not be reported when `Thread#report_on_exception=true` and `Thread#abort_on_exception=true`  

01/27/2017 06:13 AM - tagomoris (Satoshi TAGOMORI)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>nobu (Nobuyoshi Nakada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v:</td>
</tr>
<tr>
<td></td>
<td>Backport: 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE</td>
</tr>
</tbody>
</table>

**Description**

When we set true on both of `Thread#report_on_exception` and `Thread#abort_on_exception`, it just works as `Thread#abort_on_exception` and raised error will be re-raised in main thread. But in our case, main thread sometimes in begin-rescue clause, or sometimes out of begin-rescue clause. When main thread in begin-rescue clause (and main thread code may raise errors in their own business), we cannot know that threads are dead by errors.

What I expected by specifying both of `Thread#report_on_exception=true` and `Thread#abort_on_exception=true`:

- "Thread terminated with exception" is reported always (even if `Thread#abort_on_exception` is set)
- Raised error is re-raised in main thread if `Thread#abort_on_exception` is true

How do you think about this idea?

**Associated revisions**

Revision ab81cc4d - 09/19/2017 02:42 AM - nobu (Nobuyoshi Nakada)

thread.c: report then abort

- thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]

Revision 59963 - 09/19/2017 02:42 AM - nobu (Nobuyoshi Nakada)

thread.c: report then abort

- thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]

Revision 59963 - 09/19/2017 02:42 AM - nobu (Nobuyoshi Nakada)

thread.c: report then abort

- thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]

Revision 59963 - 09/19/2017 02:42 AM - nobu (Nobuyoshi Nakada)

thread.c: report then abort

- thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]

Revision 3c35067c - 03/10/2018 02:32 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59963: [Backport #13163]

```
thread.c: report then abort

* thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_4@62711 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62711 - 03/10/2018 02:32 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59963: [Backport #13163]
thread.c: report then abort

* thread.c (thread_start_func_2): report then abort on exception, if both are set. [ruby-core:79280] [Bug #13163]

History

#1 - 04/18/2017 04:20 AM - shyouhei (Shyouhei Urabe)
No objection.

#2 - 09/16/2017 02:25 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Feedback
- Description updated

Isn't this a bug?

#3 - 09/16/2017 02:47 PM - Eregon (Benoit Daloze)
+1
The patch looks good.
The message of the assert_warn should be changed though.

#4 - 09/19/2017 02:02 AM - naruse (Yui NARUSE)
- Backport set to 2.2: DONTNEED, 2.3: REQUIRED, 2.4: REQUIRED
- Assignee set to nobu (Nobuyoshi Nakada)
- Status changed from Feedback to Assigned
- Tracker changed from Misc to Bug

Hmm, indeed this seems considered a bug.
nobu (Nobuyoshi Nakada) Could you commit it?

#5 - 09/19/2017 02:42 AM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed

Applied in changeset trunk|r59963.

#6 - 10/15/2017 05:50 PM - Eregon (Benoit Daloze)
- Backport changed from 2.2: DONTNEED, 2.3: REQUIRED, 2.4: REQUIRED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED

I believe we only need a 2.4 backport, Thread#report_on_exception is only defined from Ruby 2.4+.

#7 - 03/10/2018 02:33 AM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE

ruby_2_4 r62711 merged revision(s) 59963.