

## Ruby master - Bug #12082

### Tail-calling method can't catch exception raised by tail-called method

02/18/2016 04:35 AM - rhenium (Kazuki Yamaguchi)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> ko1 (Koichi Sasada)	
<b>Target version:</b>	
<b>ruby -v:</b>	<b>Backport:</b> 2.1: WONTFIX, 2.2: DONE, 2.3: DONE
<b>Description</b>	
<p>The following code doesn't work as expected, on all versions of Ruby with tail call optimization (1.9.1 to 2.3.0).</p> <pre>def do_raise   raise "should be rescued" end options = {   tailcall_optimization: true,   trace_instruction: false, } RubyVM::InstructionSequence.compile(&lt;&lt;EOF, __FILE__, __FILE__, __LINE__, options).eval def test_rescue   return do_raise   1 + 2 rescue   :ok end EOF  p test_rescue # should print :ok, but raises "should be rescued"</pre> <p>Looks like nop instruction is (also) used to avoid this optimization (<a href="#">compile.c</a>), but when doing early return, no nop is inserted.</p> <p>I attached a (dirty) fix for this. Maybe there is a cleaner way.</p>	

#### Associated revisions

##### Revision f0226f0a - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)

compile.c: disable tco with rescue

- compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@54542 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 54542 - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)

compile.c: disable tco with rescue

- compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

##### Revision 54542 - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)

compile.c: disable tco with rescue

- compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

##### Revision 54542 - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)

compile.c: disable tco with rescue

- compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

##### Revision 54542 - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)

compile.c: disable tco with rescue

- `compile.c (iseq_optimize)`: disable tail call optimization in `rescued`, `rescue`, and `ensure` blocks. [`ruby-core:73871`] [Bug #12082]

#### Revision 6648d615 - 04/12/2016 02:51 AM - nobu (Nobuyoshi Nakada)

`compile.c`: initialize LABEL fields

- `compile.c (new_label_body)`: initialize bit fields, since `compile_data_alloc` does not clear the memory. [Bug #12082]

git-svn-id: `svn+ssh://ci.ruby-lang.org/ruby/trunk@54548 b2dd03c8-39d4-4d8f-98ff-823fe69b080e`

#### Revision 54548 - 04/12/2016 02:51 AM - nobu (Nobuyoshi Nakada)

`compile.c`: initialize LABEL fields

- `compile.c (new_label_body)`: initialize bit fields, since `compile_data_alloc` does not clear the memory. [Bug #12082]

#### Revision 54548 - 04/12/2016 02:51 AM - nobu (Nobuyoshi Nakada)

`compile.c`: initialize LABEL fields

- `compile.c (new_label_body)`: initialize bit fields, since `compile_data_alloc` does not clear the memory. [Bug #12082]

#### Revision 54548 - 04/12/2016 02:51 AM - nobu (Nobuyoshi Nakada)

`compile.c`: initialize LABEL fields

- `compile.c (new_label_body)`: initialize bit fields, since `compile_data_alloc` does not clear the memory. [Bug #12082]

#### Revision 54548 - 04/12/2016 02:51 AM - nobu (Nobuyoshi Nakada)

`compile.c`: initialize LABEL fields

- `compile.c (new_label_body)`: initialize bit fields, since `compile_data_alloc` does not clear the memory. [Bug #12082]

#### Revision bb7fa59e - 04/22/2016 09:42 AM - usa (Usaku NAKAMURA)

merge revision(s) 54542,54548: [Backport #12082]

```
* compile.c (iseq_optimize): disable tail call optimization in
rescued, rescue, and ensure blocks.
[ruby-core:73871] [Bug #12082]
```

```
* compile.c (new_label_body): initialize bit fields, since
compile_data_alloc does not clear the memory. [Bug #12082]
```

git-svn-id: `svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_2@54698 b2dd03c8-39d4-4d8f-98ff-823fe69b080e`

#### Revision 54698 - 04/22/2016 09:42 AM - usa (Usaku NAKAMURA)

merge revision(s) 54542,54548: [Backport #12082]

```
* compile.c (iseq_optimize): disable tail call optimization in
rescued, rescue, and ensure blocks.
[ruby-core:73871] [Bug #12082]
```

```
* compile.c (new_label_body): initialize bit fields, since
compile_data_alloc does not clear the memory. [Bug #12082]
```

#### Revision b8aa4100 - 04/22/2016 03:59 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 54141,54542,54548: [Backport #12082]

```
test_optimization.rb: tailcall
```

```
* test/ruby/test_optimization.rb (TestRubyOptimization.tailcall):
```

helper method to compile methods with tailcall optimization enabled.

```
* compile.c (iseq_optimize): disable tail call optimization in
rescued, rescue, and ensure blocks.
```

[`ruby-core:73871`] [Bug #12082]

```
* compile.c (new_label_body): initialize bit fields, since
compile_data_alloc does not clear the memory. [Bug #12082]
```

**Revision 54715 - 04/22/2016 03:59 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) 54141,54542,54548: [Backport #12082]

test\_optimization.rb: tailcall

```
* test/ruby/test_optimization.rb (TestRubyOptimization.tailcall):
```

helper method to compile methods with tailcall optimization enabled.

\* compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

```
* compile.c (new_label_body): initialize bit fields, since compile_data_alloc does not clear the memory. [Bug #12082]
```

**Revision cf23ad42 - 04/27/2016 01:17 PM - usa (Usaku NAKAMURA)**

- compile.c (new\_label\_body): missed backporting r54548 in r54698. this fixes randomly test failure introduced by r54698. cf. [Bug #12082]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_2@54799 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 54799 - 04/27/2016 01:17 PM - usa (Usaku NAKAMURA)**

- compile.c (new\_label\_body): missed backporting r54548 in r54698. this fixes randomly test failure introduced by r54698. cf. [Bug #12082]

## History

---

**#1 - 02/18/2016 05:46 AM - ko1 (Koichi Sasada)**

- Assignee set to ko1 (Koichi Sasada)

**#2 - 03/09/2016 09:24 AM - rhenium (Kazuki Yamaguchi)**

- File 0001-compile.c-don-t-do-tail-call-optimization-if-covered-v2.patch added

Updating my patch, because it breaks such code:

```
def errinfo
  $!
end
RubyVM::InstructionSequence.compile(<<EOF, __FILE__, __FILE__, __LINE__, tailcall_optimization: true).eval
  def test_rescue
    raise "a"
  rescue
    errinfo
  end
EOF
p test_rescue # should return a RuntimeError
```

**#3 - 03/18/2016 08:52 AM - nobu (Nobuyoshi Nakada)**

[https://github.com/ruby/ruby/compare/trunk...nobu:bug/12082-tailcall\\_inhibited\\_by\\_rescue?expand=1](https://github.com/ruby/ruby/compare/trunk...nobu:bug/12082-tailcall_inhibited_by_rescue?expand=1)

**#4 - 04/11/2016 05:57 AM - ko1 (Koichi Sasada)**

Thank you for reporting and patches.

Nobu's patch seems good. Could you commit it?

**#5 - 04/11/2016 08:43 AM - nobu (Nobuyoshi Nakada)**

- Status changed from Open to Closed

Applied in changeset r54542.

---

compile.c: disable tco with rescue

- compile.c (iseq\_optimize): disable tail call optimization in rescued, rescue, and ensure blocks. [ruby-core:73871] [Bug #12082]

**#6 - 04/22/2016 08:07 AM - usa (Usaku NAKAMURA)**

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN to 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED

**#7 - 04/22/2016 09:42 AM - usa (Usaku NAKAMURA)**

- Backport changed from 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED to 2.1: WONTFIX, 2.2: DONE, 2.3: REQUIRED

ruby\_2\_2 r54698 merged revision(s) 54542,54548.

**#8 - 04/22/2016 03:59 PM - nagachika (Tomoyuki Chikanaga)**

- Backport changed from 2.1: WONTFIX, 2.2: DONE, 2.3: REQUIRED to 2.1: WONTFIX, 2.2: DONE, 2.3: DONE

ruby\_2\_3 r54715 merged revision(s) 54141,54542,54548.

**Files**

---

0001-compile.c-don-t-do-tail-call-optimization-if-covered.patch	7.21 KB	02/18/2016	rhenium (Kazuki Yamaguchi)
0001-compile.c-don-t-do-tail-call-optimization-if-covered-v2.patch	7.49 KB	03/09/2016	rhenium (Kazuki Yamaguchi)