Ruby master - Bug #7668
set_trace_func and TracePoint don't handle exception in finish frame

01/07/2013 03:01 PM - ko1 (Koichi Sasada)

<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>ko1 (Koichi Sasada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 2.0.0dev (2013-01-07 trunk 38719)</td>
</tr>
<tr>
<td>Backport:</td>
<td></td>
</tr>
</tbody>
</table>

### Description

The following code should trap exception correctly (should output ":ok"):

```ruby
####
def m
  a = 1
  b = 2
  c = 3
  raise
end

trace = TracePoint.new{|tp|
  p tp
  raise # if tp.event == :c_return # if tp.event == :b_return
}

begin
  trace.enable{
    m
  }
rescue => e
  p :ok
end
####
```

But it outputs same event hook infinite:

```ruby
#TracePoint:b_call@/home/ko1/src/ruby/trunk/test.rb:15
#TracePoint:b_return@/home/ko1/src/ruby/trunk/test.rb:15
#
#
#
...
```

It is a bug.

This patch solve this issue:

### Index: vm_trace.c

```c
--- vm_trace.c (revision 38718)
+++ vm_trace.c (working copy)
@@ -316,7 +316,12 @@ rb_threadptr_exec_event_hooks_orig(rb_tr
          th->vm->trace_running--;

         if (state) {
               if (pop_p) th->cfp = RUBY_VM_PREVIOUS_CONTROL_FRAME(th->cfp);
               if (pop_p) {
                 if (VM_FRAME_TYPE_FINISH_P(th->cfp)) {
                   th->tag = th->tag->prev;
```
th->cfp = RUBY_VM_PREVIOUS_CONTROL_FRAME(th->cfp);
TH_JUMP_TAG(th, state); th->state = outer_state;

Output:
# TracePoint: b_call@../trunk/test.rb:15
# TracePoint: b_return@../trunk/test.rb:15
:ok

Associated revisions
Revision b8e6fd6f - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

Revision 38721 - 01/07/2013 06:24 AM - ko1 (Koichi Sasada)
- vm_trace.c (rb_threadptr_exec_event_hooks_orig): pop tag before JUMP_TAG() if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- test/ruby/test_settracefunc.rb: add a test for above.

History
#1 - 01/07/2013 03:24 PM - ko1 (Koichi Sasada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r38721.
Koichi, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.
- `vm_trace.c (rb_threadptr_exec_event_hooks_orig)`: pop tag before `JUMP_TAG()` if frame is 'finish' frame. Without this patch, there is an inconsistency between control frame stack and tags stack. [Bug #7668]
- `test/ruby/test_settracefunc.rb`: add a test for above.