Ruby trunk - Bug #11335

`ruby -r debug` catchpoint problem

07/05/2015 11:27 AM - sigsys (Math Ieu)

Status:	Open		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.1.6p336 (2015-04-13 revision 50298) [i386-freebsd10]	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

With a test-debug.rb like this:

```
raise 'test'
```

And starting the debugger like this:

```
ruby -r debug test-debug.rb
```

By default, the catchpoint is StandardError, but it doesn't work:

```
test-debug.rb:1:raise 'test'
(rdb:1) catch
Catchpoint StandardError.
(rdb:1) c
test-debug.rb:1:in `<main>': test (RuntimeError)
```

And the debugger exits without catching the exception.

But, by setting the catchpoint to NilClass (or one of its ancestors), then it works:

```
test-debug.rb:1:raise "test"
(rdb:1) catch NilClass
Set catchpoint NilClass.
(rdb:1) c
test-debug.rb:1: `' (NilClass)
        from test-debug.rb:1:in `<main>'
test-debug.rb:1:raise "test"
```

And the debugger does not exit and allows further debugging.

By looking at lib/debug.rb, it looks like that the set_trace_func callback it sets assumes that $! will be set to the exception to be raised when a 'raise' event occurs. But it is not the case, $! seems to always be nil.

History

#1 - 12/07/2018 08:32 AM - oleynikov (Alexander Oleynikov)

I was able to reproduce the issue on newer Ruby.

```
$ ruby -v
ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-openbsd]
```

sigsys (Math Ieu) wrote:

> By looking at lib/debug.rb, it looks like that the set_trace_func callback it sets assumes that $! will be set to the exception to be raised when a 'raise' event occurs. But it is not the case, $! seems to always be nil.

Exactly. When set_trace_func handler proc gets executed with a 'raise' event, the $! is still nil.
DEBUGGER__::Context#excn_handle wants to find a class of the raised exception in $!, but always finds nil there.

We could fix that by replacing Kernel#set_trace_func with TracePoint. It is the recommended way according to the docs, where set_trace_func since Ruby 2.1 contains a note, saying "this method is obsolete, please use TracePoint instead". Instead of $! we should use TracePoint#raised_exception to get correct results.

I'm a bit afraid to work on this issue myself, because I found no specs for the debugger and I can accidentally break something without noticing. But if nobody else is interested in doing this, I will try.