Add new set_trace_func events "b-call", "b-return"

06/26/2012 04:09 PM - ko1 (Koichi Sasada)

Status: Rejected
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version:

Description

=begin
= Abstract
Add new events "b-call", "b-return" for set_trace_func().
When block is invoked, then "b-call" is called.
When block is finished, then "b-return" is called.

= Background

set_trace_func() has several events:

  • "line"
  • "call", "return"
  • "c-call", "c-return"
  • "class", "end"
  • "raise"

With above events, we can trace execution. However, we can't trace block invocation because there are no events related to block invocation and return from block.

For example, performance profiler using set_trace_func() has problem in such case:

def m1
  yield
end

def m2
  m1 do
    # long time process
  end
end

A method m1 doesn't consume time, but m2 consume long time (within block).
However, the profiler detects that the method `m1' consumes long time.

= Proposal

Add "b-call" and "b-return" events for set_trace_func. "b-call" is for block invocation. "b-return" is for returning from block.

Parameters:

  • event: "b-call" or "b-return"
  • file: filename of block
  • line: start/end line number of block
  • id: block owner method id
  • binding: binding of block
  • klass: block owner class

= Concerns

  • This proposal breaks 100% compatibility. It seems problem (*1).
  • It will make some runtime overhead (slow down).
  • parameters id and klass is appropriate?
Making an alternative API which can replace set_trace_func() (correctly speaking, set_trace_func() can be implemented with this new API) is another proposal.

```ruby
# as usual, the method name is temporary name
set_trace_func2{|info|
  info.event
}
```

History

#1 - 06/26/2012 04:18 PM - ko1 (Koichi Sasada)
- Description updated

#2 - 06/26/2012 11:32 PM - trans (Thomas Sawyer)

Making an alternative API which can replace set_trace_func() (correctly speaking, set_trace_func() can be implemented with this new API) is another proposal.

I assume referring to [ruby-core:5528](http://ruby-core.org/mail-archive/index.html?thread_id=5528)

#3 - 06/27/2012 10:29 AM - ko1 (Koichi Sasada)
(2012/06/26 23:32), trans (Thomas Sawyer) wrote:

I assume referring to [ruby-core:5528](http://ruby-core.org/mail-archive/index.html?thread_id=5528)

Sound interesting.

Can I see the reference of TracePoint? I found README.rdoc and lib/tracepoint.rb, but I can't find the specification what methods tp (block parameter: TracePoint.trace(type) do |tp| ... end) has.

```ruby
// SASADA Koichi at atdot dot net
```

#4 - 06/27/2012 09:54 PM - trans (Thomas Sawyer)
This should help. I added some YARD docs. See [http://rubydoc.info/github/rubyworks/tracepoint/master/frames](http://rubydoc.info/github/rubyworks/tracepoint/master/frames)

#5 - 06/29/2012 03:53 PM - ko1 (Koichi Sasada)
(2012/06/27 21:54), trans (Thomas Sawyer) wrote:

This should help. I added some YARD docs. See [http://rubydoc.info/github/rubyworks/tracepoint/master/frames](http://rubydoc.info/github/rubyworks/tracepoint/master/frames)

Thanks. I understand the API of TracePoint.

How about to close this ticket and 5528, and re-new ticket about TracePoint API for Ruby 2.0?

My rough idea:

- Add TracePoint class
  - The class name should be considered.
    - I have no idea about it.
    - ::TracePoint or RubyVM::TracePoint or Thread::TracePoint?

- Add event-separated hooks
  - TracePoint.trace(event_name) { ... } seems good.
  - But how to choose Thread local or system global?
    - Now, we have Kernel#set_trace_func and Thread#set_trace_func
  - One idea is Kernel#trace_point(event_name) and Thread#trace_point(event_name)
• Add basic methods (attributes)
  • binding
  • event # event signatures should be symbol, I think.
    • :call, :return
    • :ccall, :creturn
    • :bcall, :breturn # new!
    • :class, :and
    • :raise
    • :line
  • file
  • line
  • klass/id are not in TracePoint. No need to have?

• I think "call?" or other methods are not needed because we can define in Ruby. Are they indispensable?

--
// SASADA Koichi at atdot dot net

#6 - 07/14/2012 06:38 PM - mame (Yusuke Endoh)
- Status changed from Open to Assigned

#7 - 07/25/2012 07:06 PM - ko1 (Koichi Sasada)
- Status changed from Assigned to Rejected

Nobody has interest about it.
I close (reject) this ticket and make new ticket of
https://bugs.ruby-lang.org/issues/6649#note-5,
http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/45949
with implementation.

If you have any idea, please teach me.