

Ruby trunk - Bug #3841

RubyVM::InstructionSequence.to_a() and disasm() do not work properly for "for <var> in <list>"

09/16/2010 10:27 PM - s_p_oneil (Sean O'Neil)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:	2.6	
ruby -v:	ruby 1.9.2p0 (2010-08-18 revision 29036) [i386-mswin32_90]	Backport: 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

I have been playing with the concept of caching compiled Ruby instruction sequences and reloading them later. For now the simplest way to do it is to use RubyVM::InstructionSequence.to_a() and Marshal.dump() to convert the resulting array to a smaller binary byte sequence. While testing this, I found that some Ruby scripts worked fine when an instruction sequence was reloaded, but others broke.

As a test, I decided to try compiling irb.rb and its dependencies. When I try to reload the compiled sequences, the first file that breaks is irb/extend-command.rb. I tracked it down to the way "for in" code block and parameters are handled, but I am not sure how to fix it. When I converted each "for in" loop to ".each {||", it started working perfectly. Here's an example of how the disassembly differs between the two loops:

```
def self.install_extend_commands
  @EXTEND_COMMANDS.each {|args|
    def_extend_command(*args)
  }
end
== disasm: <RubyVM::InstructionSequence:block in install_extend_commands@irb/extend-command.rb>
== catch table
| catch type: redo   st: 0000 ed: 0012 sp: 0000 cont: 0000
| catch type: next   st: 0000 ed: 0012 sp: 0000 cont: 0012
|-----|
local table (size: 2, argc: 1 [opts: 0, rest: -1, post: 0, block: -1] s3)
[ 2] args<Arg>
0000 trace           1                               ( 110)
0002 putnil
0003 getdynamic      args, 0
0006 send            :def_extend_command, 1, nil, 10, <ic:0>
0012 leave

def self.install_extend_commands
  for args in @EXTEND_COMMANDS
    def_extend_command(*args)
  end
end
== disasm: <RubyVM::InstructionSequence:block in install_extend_commands@irb/extend-command.rb>
== catch table
| catch type: redo   st: 0005 ed: 0016 sp: 0000 cont: 0005
| catch type: next   st: 0005 ed: 0016 sp: 0000 cont: 0016
|-----|
local table (size: 2, argc: 1 [opts: 0, rest: -1, post: 0, block: -1] s3)
[ 2] ?<Arg>
0000 getdynamic      *, 0                               ( 206)
0003 setlocal        args                               ( 204)
0005 trace           1                               ( 205)
0007 putnil
0008 getlocal        args
0010 send            :def_extend_command, 1, nil, 10, <ic:0>
0016 leave
```

In this example, rb_id2name() returns NULL for the name of the argument passed to the code block. That is why it puts a "?" in the local table and a "*" in the getdynamic command. The to_a() method has a similar problem, and it appears to put a number there instead. When I use the load() method to convert the array back into an instruction sequence and try to execute it, the code block

ends up receiving a number as a parameter instead of the entry in the list it is supposed to get.

I imagine this is not a high priority for the Ruby core team, and I don't mind putting in some work to fix it. However, it looks like it would take me a while to figure out, so I would appreciate any tips you could give me to point me in the right direction.

Thank you,
Sean O'Neil

History

#1 - 09/26/2010 04:36 AM - shyouhei (Shyouhei Urabe)

- Status changed from *Open* to *Assigned*
- Assignee set to *ko1* (Koichi Sasada)

=begin

=end

#2 - 06/11/2011 03:18 PM - ko1 (Koichi Sasada)

- Priority changed from *Normal* to *3*
- Target version set to *2.0.0*

If I have a time, I'll check it.

#3 - 11/26/2012 09:09 AM - ko1 (Koichi Sasada)

- Target version changed from *2.0.0* to *2.6*

#4 - 01/31/2017 08:40 AM - ko1 (Koichi Sasada)

- Status changed from *Assigned* to *Closed*
- Description updated

I'm not sure who fix this issue, but 2.5dev doesn't have this issue.