

Ruby trunk - Bug #7805

ruby 2.0rc2 core on solaris

02/09/2013 04:04 AM - groenveld@acm.org (John Groenveld)

Status:	Closed	
Priority:	Normal	
Assignee:	nobu (Nobuyoshi Nakada)	
Target version:	2.6	
ruby -v:	ruby 2.0.0dev (2013-02-08 trunk 39161) [i386-solaris2.10]	Backport:

Description

```
$ env PATH=/usr/bin:/usr/sbin:/usr/ccs/bin:/opt/solarisstudio12.3/bin:/tmp/bin \
CC=cc CFLAGS="-m64 -O3" CXX=CC CXXFLAGS=-m64 CPPFLAGS="-I/opt/apache2/yaml/include -I/usr/sfw/include" \
LDFLAGS="-m64 -L/usr/sfw/lib/64 -R/usr/sfw/lib/64 -L/opt/apache2/yaml/lib -R/opt/apache2/yaml/lib" \
MAKE=gmake ./configure --prefix=/opt/apache2/ruby-1.9.3 --enable-shared --without-gcc
```

```
Generating RI format into /tmp/ruby-2.0.0-rc2/.ext/rdoc...
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:888: [BUG] Segmentation fault
ruby 2.0.0dev (2013-02-08 trunk 39161) [i386-solaris2.10]
```

```
-- Control frame information -----
c:0016 p:---- s:0057 e:000056 CFUNC :dump
c:0015 p:0126 s:0053 e:000052 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:888
c:0014 p:0011 s:0045 e:000044 BLOCK /tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:750
c:0013 p:0008 s:0042 e:000041 BLOCK /tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb:710 [FINISH]
c:0012 p:---- s:0039 e:000038 CFUNC :each
c:0011 p:0029 s:0036 e:000035 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb:710
c:0010 p:0016 s:0033 e:000032 BLOCK /tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:749 [FINISH]
c:0009 p:---- s:0030 e:000029 CFUNC :each
c:0008 p:0015 s:0027 e:000026 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:746
c:0007 p:0009 s:0024 e:000023 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/generator/ri.rb:26
c:0006 p:0057 s:0021 e:000020 BLOCK /tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:526 [FINISH]
c:0005 p:---- s:0019 e:000018 CFUNC :chdir
c:0004 p:0018 s:0015 e:000014 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:521
c:0003 p:0354 s:0012 e:000011 METHOD /tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:504
c:0002 p:0049 s:0006 E:0002f8 EVAL ./bin/rdoc:20 [FINISH]
c:0001 p:0000 s:0002 E:0007a8 TOP [FINISH]
```

```
-- Ruby level backtrace information -----
./bin/rdoc:20:in <main>'
/tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:504:in document'
/tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:521:in generate'
/tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:521:in chdir'
/tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb:526:in block in generate'
/tmp/ruby-2.0.0-rc2/lib/rdoc/generator/ri.rb:26:ingenerate'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:746:in save'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:746:ineach'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:749:in block in save'
/tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb:710:ineach_method'
/tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb:710:in each'
/tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb:710:inblock in each_method'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:750:in block (2 levels) in save'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:888:insave_method'
/tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb:888:in `dump'
```

```
-- Other runtime information -----
```

- Loaded script: ./bin/rdoc
- Loaded features:
 - 0 enumerator.so

- 1 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/encdb.so
- 2 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/trans/transdb.so
- 3 /tmp/ruby-2.0.0-rc2/lib/rdoc.rb
- 4 /tmp/ruby-2.0.0-rc2/lib/find.rb
- 5 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/etc.so
- 6 /tmp/ruby-2.0.0-rc2/lib/fileutils.rb
- 7 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/pathname.so
- 8 /tmp/ruby-2.0.0-rc2/.ext/common/pathname.rb
- 9 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/date_core.so
- 10 /tmp/ruby-2.0.0-rc2/.ext/common/date/format.rb
- 11 /tmp/ruby-2.0.0-rc2/.ext/common/date.rb
- 12 /tmp/ruby-2.0.0-rc2/lib/time.rb
- 13 /tmp/ruby-2.0.0-rc2/rbconfig.rb
- 14 /tmp/ruby-2.0.0-rc2/lib/rubygems/compatibility.rb
- 15 /tmp/ruby-2.0.0-rc2/lib/rubygems/defaults.rb
- 16 /tmp/ruby-2.0.0-rc2/lib/rubygems/deprecate.rb
- 17 /tmp/ruby-2.0.0-rc2/lib/rubygems/errors.rb
- 18 /tmp/ruby-2.0.0-rc2/lib/rubygems/version.rb
- 19 /tmp/ruby-2.0.0-rc2/lib/rubygems/requirement.rb
- 20 /tmp/ruby-2.0.0-rc2/lib/rubygems/platform.rb
- 21 /tmp/ruby-2.0.0-rc2/lib/rubygems/specification.rb
- 22 /tmp/ruby-2.0.0-rc2/lib/rubygems/exceptions.rb
- 23 /tmp/ruby-2.0.0-rc2/lib/rubygems/core_ext/kernel_gem.rb
- 24 /tmp/ruby-2.0.0-rc2/lib/rubygems/core_ext/kernel_require.rb
- 25 /tmp/ruby-2.0.0-rc2/lib/rubygems.rb
- 26 /tmp/ruby-2.0.0-rc2/lib/rubygems/path_support.rb
- 27 /tmp/ruby-2.0.0-rc2/lib/cgi/util.rb
- 28 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/strscan.so
- 29 /tmp/ruby-2.0.0-rc2/lib/erb.rb
- 30 /tmp/ruby-2.0.0-rc2/lib/rdoc/generator.rb
- 31 /tmp/ruby-2.0.0-rc2/lib/rubygems/dependency.rb
- 32 /tmp/ruby-2.0.0-rc2/.ext/common/json/version.rb
- 33 /tmp/ruby-2.0.0-rc2/lib/ostruct.rb
- 34 /tmp/ruby-2.0.0-rc2/.ext/common/json/generic_object.rb
- 35 /tmp/ruby-2.0.0-rc2/.ext/common/json/common.rb
- 36 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/utf_16be.so
- 37 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/utf_16le.so
- 38 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/utf_32be.so
- 39 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/enc/utf_32le.so
- 40 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/json/ext/parser.so
- 41 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/json/ext/generator.so
- 42 /tmp/ruby-2.0.0-rc2/.ext/common/json/ext.rb
- 43 /tmp/ruby-2.0.0-rc2/.ext/common/json.rb
- 44 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup.rb
- 45 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/formatter.rb
- 46 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/to_joined_paragraph.rb
- 47 /tmp/ruby-2.0.0-rc2/lib/rdoc/markdown/entities.rb
- 48 /tmp/ruby-2.0.0-rc2/lib/rdoc/markdown/literals_1_9.rb
- 49 /tmp/ruby-2.0.0-rc2/lib/rdoc/markdown.rb
- 50 /tmp/ruby-2.0.0-rc2/lib/rdoc/rd.rb
- 51 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/parser.rb
- 52 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/pre_process.rb
- 53 /tmp/ruby-2.0.0-rc2/lib/rdoc/tom_doc.rb
- 54 /tmp/ruby-2.0.0-rc2/lib/rdoc/text.rb
- 55 /tmp/ruby-2.0.0-rc2/lib/rdoc/code_object.rb
- 56 /tmp/ruby-2.0.0-rc2/lib/rdoc/method_attr.rb
- 57 /tmp/ruby-2.0.0-rc2/lib/cgi/core.rb
- 58 /tmp/ruby-2.0.0-rc2/lib/cgi/cookie.rb
- 59 /tmp/ruby-2.0.0-rc2/lib/cgi.rb
- 60 /tmp/ruby-2.0.0-rc2/lib/rdoc/context.rb
- 61 /tmp/ruby-2.0.0-rc2/lib/rdoc/class_module.rb
- 62 /tmp/ruby-2.0.0-rc2/lib/rdoc/context/section.rb
- 63 /tmp/ruby-2.0.0-rc2/lib/rdoc/top_level.rb
- 64 /tmp/ruby-2.0.0-rc2/lib/rdoc/generator/markup.rb
- 65 /tmp/ruby-2.0.0-rc2/lib/rdoc/generator/darkfish.rb
- 66 /tmp/ruby-2.0.0-rc2/lib/rdoc/generator/ri.rb
- 67 /tmp/ruby-2.0.0-rc2/lib/rdoc/rdoc.rb

```

68 /tmp/ruby-2.0.0-rc2/lib/rdoc/store.rb
69 /tmp/ruby-2.0.0-rc2/lib/optparse.rb
70 /tmp/ruby-2.0.0-rc2/lib/rdoc/options.rb
71 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/text.rb
72 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/simple.rb
73 /tmp/ruby-2.0.0-rc2/lib/rdoc/tsort.rb
74 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/c.rb
75 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/changelog.rb
76 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/markdown.rb
77 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/rd.rb
78 /tmp/ruby-2.0.0-rc2/lib/rdoc/ruby_token.rb
79 /tmp/ruby-2.0.0-rc2/lib/rdoc/token_stream.rb
80 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/ruby_tools.rb
81 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser/ruby.rb
82 /tmp/ruby-2.0.0-rc2/lib/rdoc/parser.rb
83 /tmp/ruby-2.0.0-rc2/lib/rdoc/ri.rb
84 /tmp/ruby-2.0.0-rc2/lib/rdoc/ri/paths.rb
85 /tmp/ruby-2.0.0-rc2/lib/rdoc/stats.rb
86 /tmp/ruby-2.0.0-rc2/lib/rdoc/stats/quiet.rb
87 /tmp/ruby-2.0.0-rc2/lib/rdoc/stats/normal.rb
88 /tmp/ruby-2.0.0-rc2/lib/rdoc/encoding.rb
89 /tmp/ruby-2.0.0-rc2/lib/rdoc/comment.rb
90 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/document.rb
91 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/heading.rb
92 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/blank_line.rb
93 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/list.rb
94 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/raw.rb
95 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/paragraph.rb
96 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/list_item.rb
97 /tmp/ruby-2.0.0-rc2/lib/rdoc/known_classes.rb
98 /tmp/ruby-2.0.0-rc2/lib/rdoc/normal_class.rb
99 /tmp/ruby-2.0.0-rc2/lib/rdoc/any_method.rb
100 /tmp/ruby-2.0.0-rc2/lib/rdoc/include.rb
101 /tmp/ruby-2.0.0-rc2/lib/rdoc/alias.rb
102 /tmp/ruby-2.0.0-rc2/lib/rdoc/normal_module.rb
103 /tmp/ruby-2.0.0-rc2/lib/rdoc/constant.rb
104 /tmp/ruby-2.0.0-rc2/lib/rdoc/attr.rb
105 /tmp/ruby-2.0.0-rc2/lib/e2mmap.rb
106 /tmp/ruby-2.0.0-rc2/lib/irb/output-method.rb
107 /tmp/ruby-2.0.0-rc2/lib/irb/notifier.rb
108 /tmp/ruby-2.0.0-rc2/lib/irb/slex.rb
109 /tmp/ruby-2.0.0-rc2/.ext/i386-solaris2.10/stringio.so
110 /tmp/ruby-2.0.0-rc2/lib/rdoc/ruby_lex.rb
111 /tmp/ruby-2.0.0-rc2/lib/rdoc/require.rb
112 /tmp/ruby-2.0.0-rc2/lib/rdoc/extend.rb
113 /tmp/ruby-2.0.0-rc2/lib/rdoc/ghost_method.rb
114 /tmp/ruby-2.0.0-rc2/lib/rdoc/meta_method.rb
115 /tmp/ruby-2.0.0-rc2/lib/rdoc/markup/verbatim.rb

```

[NOTE]

You may have encountered a bug in the Ruby interpreter or extension libraries.
Bug reports are welcome.

```

$ pstack .ext/rdoc/core
core '.ext/rdoc/core' of 4987: ./ruby --disable-gems ./bin/rdoc --root . --page-dir ./doc --encoding=
----- lwp# 1 / thread# 1 -----
ffffd7ffef1351a _lwp_kill () + a
ffffd7fffeb81b9 raise () + 19
ffffd7ffef96b80 abort () + 90
ffffd7fff0a6d90 rb_bug () + c0
ffffd7fff1643ee sigsegv () + 4e
ffffd7ffef0dd16 __sighndlr () + 6
ffffd7ffef025e2 call_user_handler () + 252
ffffd7ffef0280e sigacthandler (b, 5cfb0, 5cfb50) + ee
--- called from signal handler with signal 11 (SIGSEGV) ---
ffffd7fff16dc66 st_lookup () + 16

```

```

fffffd7fff0dcada w_class () + 2a
fffffd7fff0ddd15 w_object () + ce5
fffffd7fff0ddfe9 marshal_dump () + 199
fffffd7fff1c8bad vm_call_cfunc_with_frame () + 34d
fffffd7fff1cdd88 vm_exec_core () + 2cc8
fffffd7fff1da39f vm_exec () + b0f
fffffd7fff1d8a80 invoke_block_from_c () + 4b0
fffffd7fff1d4f33 rb_yield () + 73
fffffd7fff073947 rb_ary_each () + 77
fffffd7fff1c8bad vm_call_cfunc_with_frame () + 34d
fffffd7fff1cde16 vm_exec_core () + 2d56
fffffd7fff1da39f vm_exec () + b0f
fffffd7fff1d8a80 invoke_block_from_c () + 4b0
fffffd7fff1d4f33 rb_yield () + 73
fffffd7fff073947 rb_ary_each () + 77
fffffd7fff1c8bad vm_call_cfunc_with_frame () + 34d
fffffd7fff1c9eb4 vm_call_method () + 3c4
fffffd7fff1cde16 vm_exec_core () + 2d56
fffffd7fff1da39f vm_exec () + b0f
fffffd7fff1d8a80 invoke_block_from_c () + 4b0
fffffd7fff1d4f33 rb_yield () + 73
fffffd7fff0acd09 rb_ensure () + 89
fffffd7fff099dd8 dir_s_chdir () + 118
fffffd7fff1c8bad vm_call_cfunc_with_frame () + 34d
fffffd7fff1c9eb4 vm_call_method () + 3c4
fffffd7fff1cde16 vm_exec_core () + 2d56
fffffd7fff1da39f vm_exec () + b0f
fffffd7fff0abaf5 ruby_exec_internal () + 95
fffffd7fff0abc0e ruby_exec_node () + 1e
fffffd7fff0abbd4 ruby_run_node () + 24
0000000000400dcc main () + 4c
0000000000400c1b ???????? ()
----- lwp# 2 / thread# 2 -----
fffffd7ffef1305a __pollsys () + a
fffffd7ffe9b98b4 pselect () + 154
fffffd7ffe9b982 select () + 72
fffffd7fff1e23ac thread_timer () + ac
fffffd7ffef0d9db _thr_setup () + 5b
fffffd7ffef0dc10 _lwp_start ()

```

Related issues:

Related to Ruby trunk - Bug #9523: marshal_dump and callcc causes SEGV

Closed

02/16/2014

Associated revisions

Revision 82b8467e - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39860 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 39860 - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

Revision 39860 - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

Revision 39860 - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

Revision 39860 - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

Revision 39860 - 03/21/2013 01:50 PM - ngoto (Naohisa Goto)

- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert r38174. [ruby-core:52042] [Bug #7805]

Revision 71208c02 - 04/11/2013 07:23 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 39860: [Backport #8150]

```
* marshal.c (marshal_dump, marshal_load): workaround for segv on
Intel Solaris compiled with Oracle SolarisStudio 12.3.
Partly revert r38174. [ruby-core:52042] [Bug #7805]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_0_0@40257 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 08252d18 - 02/20/2014 11:45 PM - normal

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45064 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45064 - 02/20/2014 11:45 PM - normal

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

Revision 45064 - 02/20/2014 11:45 PM - normalperson (Eric Wong)

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

Revision 45064 - 02/20/2014 11:45 PM - normal

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

Revision 45064 - 02/20/2014 11:45 PM - normal

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

Revision 45064 - 02/20/2014 11:45 PM - normal

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [ruby-core:60816] [Bug #7805]

History

#1 - 02/09/2013 11:48 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to ngoto (Naohisa Goto)

#2 - 02/10/2013 10:34 PM - ngoto (Naohisa Goto)

I couldn't reproduce it on SPARC Solaris10.

Please show config.log, .ext/include/i386-solaris2.10/ruby/config.h, and all messages shown in the screen.

#3 - 02/10/2013 10:35 PM - ngoto (Naohisa Goto)

- Status changed from Assigned to Feedback

#4 - 02/19/2013 01:12 AM - mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 2.6

#5 - 02/22/2013 05:03 PM - ngoto (Naohisa Goto)

- File marshal-c-volatile.patch added

```
--- called from signal handler with signal 11 (SIGSEGV) ---
ffffd7fff16dc66 st_lookup () + 16
ffffd7fff0dcada w_class () + 2a
ffffd7fff0ddd15 w_object () + ce5
ffffd7fff0ddf9 marshal_dump () + 199
```

This may be re-occurrence of [Bug #7591] which is a GC issue.
How about the attached workaround patch?

#6 - 02/22/2013 05:12 PM - nobu (Nobuyoshi Nakada)

It means RB_GC_GUARD() doesn't work well with Solaris CC?

#7 - 02/22/2013 05:27 PM - ngoto (Naohisa Goto)

Sometimes RB_GC_GUARD() doesn't work well with Oracle SolarisStudio cc.
[Bug #5762] is another example.

#8 - 02/27/2013 02:40 AM - groenveld@acm.org (John Groenveld)

ngoto (Naohisa Goto) wrote:

```
--- called from signal handler with signal 11 (SIGSEGV) ---
ffffd7fff16dc66 st_lookup () + 16
ffffd7fff0dcada w_class () + 2a
ffffd7fff0ddd15 w_object () + ce5
ffffd7fff0ddf9 marshal_dump () + 199
```

This may be re-occurrence of [Bug #7591] which is a GC issue.
How about the attached workaround patch?

Resolves the core with Solaris Studio, thank you.

John

groenveld@acm.org

#9 - 02/27/2013 12:40 PM - kosaki (Motohiro KOSAKI)

How about to add a pragma of optimization off?
e.g. #pragma opt 0 (rb_gc_guarded_ptr)

Don't work?

#10 - 03/09/2013 02:03 PM - groenveld@acm.org (John Groenveld)

kosaki (Motohiro KOSAKI) wrote:

How about to add a pragma of optimization off?
e.g. #pragma opt 0 (rb_gc_guarded_ptr)

Don't work?

No.
John
groenveld@acm.org

#11 - 03/21/2013 10:50 PM - ngoto (Naohisa Goto)

- Status changed from Feedback to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r39860](#).

John, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

-
- marshal.c (marshal_dump, marshal_load): workaround for segv on Intel Solaris compiled with Oracle SolarisStudio 12.3. Partly revert [r38174](#). [Bug [#7805](#)]

#12 - 02/14/2014 10:11 AM - normalperson (Eric Wong)

The following should work without RB_GC_GUARD nor volatile.

I plan to commit unless there are objections, I see no possible

way for a compiler to mess things up:

```
--- a/marshal.c
+++ b/marshal.c
@@ -950,7 +950,7 @@ marshal_dump(int argc, VALUE *argv)
     VALUE obj, port, a1, a2;
     int limit = -1;
     struct dump_arg *arg;
-    volatile VALUE wrapper;
+    VALUE wrapper; /* used to avoid memory leak in case of exception */

     port = Qnil;
     rb_scan_args(argc, argv, "12", &obj, &a1, &a2);
@@ -964,7 +964,7 @@ marshal_dump(int argc, VALUE *argv)
     else if (NIL_P(a1)) io_needed();
     else port = a1;
     }
-    RB_GC_GUARD(wrapper) = TypedData_Make_Struct(rb_cData, struct dump_arg, &dump_arg_data, arg);
+    wrapper = TypedData_Make_Struct(rb_cData, struct dump_arg, &dump_arg_data, arg);
     arg->dest = 0;
     arg->symbols = st_init_numtable();
     arg->data = st_init_numtable();
@@ -993,8 +993,8 @@ marshal_dump(int argc, VALUE *argv)
     rb_io_write(arg->dest, arg->str);
     rb_str_resize(arg->str, 0);
     }
-    clear_dump_arg(arg);
-    RB_GC_GUARD(wrapper);
+    free_dump_arg(arg);
+    rb_gc_force_recycle(wrapper); /* also guards from premature GC */

     return port;
 }
@@ -1957,7 +1957,7 @@ marshal_load(int argc, VALUE *argv)
     VALUE port, proc;
     int major, minor, infection = 0;
     VALUE v;
-    volatile VALUE wrapper;
+    VALUE wrapper; /* used to avoid memory leak in case of exception */
     struct load_arg *arg;

     rb_scan_args(argc, argv, "11", &port, &proc);
@@ -1973,7 +1973,7 @@ marshal_load(int argc, VALUE *argv)
     else {
     io_needed();
     }
-    RB_GC_GUARD(wrapper) = TypedData_Make_Struct(rb_cData, struct load_arg, &load_arg_data, arg);
+    wrapper = TypedData_Make_Struct(rb_cData, struct load_arg, &load_arg_data, arg);
     arg->infection = infection;
```

```

    arg->src = port;
    arg->offset = 0;
@@ -2004,8 +2004,8 @@ marshal_load(int argc, VALUE *argv)

    if (!NIL_P(proc)) arg->proc = proc;
    v = r_object(arg);
-   clear_load_arg(arg);
-   RB_GC_GUARD(wrapper);
+   free_load_arg(arg);
+   rb_gc_force_recycle(wrapper); /* also guards from premature GC */

    return v;
}

```

<http://bogomips.org/ruby.git/patch?id=e9f200e99>

The following changes since commit 4d33c0e965a1ccce82f7e26a0ef21fef6bef3d2b:

- vm_inshelper.c (vm_call_method): should check ci->me->flag of a refining method in case the method is private. [Bug #9452] (2014-02-13 14:44:41 +0000)

are available in the git repository at:

git://80x24.org/ruby.git marshal-force_recycle

for you to fetch changes up to e9f200e99bc58894971f51b41825b8fac8e28572:

marshal.c: use rb_gc_force_recycle for GC-safety (2014-02-14 09:46:54 +0000)

Eric Wong (1):

marshal.c: use rb_gc_force_recycle for GC-safety

marshal.c | 16 ++++++-----

1 file changed, 8 insertions(+), 8 deletions(-)

#13 - 02/18/2014 07:12 AM - ngoto (Naohisa Goto)

- Status changed from Closed to Open

Please completely revert [r44994](#) and [r45025](#).
[r45025](#) partly reverts [r44494](#) and cause regression.

#14 - 02/18/2014 07:13 AM - ngoto (Naohisa Goto)

- Assignee changed from ngoto (Naohisa Goto) to nobu (Nobuyoshi Nakada)

#15 - 02/18/2014 07:19 AM - ngoto (Naohisa Goto)

- Related to Bug #9523: marshal_dump and callcc causes SEGV added

#16 - 02/18/2014 07:22 AM - normalperson (Eric Wong)

Sorry about the breakage. Instead of reverting completely and using volatile again, we should try to make RB_GC_GUARD more correct/robust.

#17 - 02/18/2014 08:12 AM - normalperson (Eric Wong)

Can you please try the following patch?

This is hopefully robust enough for all future compilers:

<http://bogomips.org/ruby.git/patch?id=1b5d3c0b9d>

```

--- a/gc.c
+++ b/gc.c
@@ -88,10 +88,14 @@
 #define rb_setjmp(env) RUBY_SETJMP(env)
 #define rb_jump_buf rb_jumpbuf_t

-#if defined(HAVE_RB_GC_GUARDED_PTR) && HAVE_RB_GC_GUARDED_PTR
+#if defined(HAVE_RB_GC_GUARDED_PTR_VAL) && HAVE_RB_GC_GUARDED_PTR_VAL
+/* trick the compiler into thinking a external signal handler uses this */
+volatile VALUE rb_gc_guarded_val;
volatile VALUE *

```



```

-rb_gc_guarded_ptr(volatile VALUE *ptr)
+rb_gc_guarded_ptr_val(volatile VALUE *ptr, VALUE val)
{
+   rb_gc_guarded_val = val;
+
+   return ptr;
}
#endif
diff --git a/include/ruby/ruby.h b/include/ruby/ruby.h
index 55ea252..abd4b4b 100644
--- a/include/ruby/ruby.h
+++ b/include/ruby/ruby.h
@@ -515,12 +515,16 @@ static inline int rb_type(VALUE obj);
+static inline volatile VALUE *rb_gc_guarded_ptr(volatile VALUE *ptr) {return ptr;}
+#pragma optimize("", on)
+else
-volatile VALUE *rb_gc_guarded_ptr(volatile VALUE *ptr);
-#define HAVE_RB_GC_GUARDED_PTR 1
+volatile VALUE *rb_gc_guarded_ptr_val(volatile VALUE *ptr, VALUE val);
+#define HAVE_RB_GC_GUARDED_PTR_VAL 1
+#define RB_GC_GUARD(v) (*rb_gc_guarded_ptr_val(&(v), (v)))
+endif
+define RB_GC_GUARD_PTR(ptr) rb_gc_guarded_ptr(ptr)
+endif
+
+ifndef RB_GC_GUARD
+define RB_GC_GUARD(v) (*RB_GC_GUARD_PTR(&(v)))
+endif

#ifdef __GNUC__
#define RB_UNUSED_VAR(x) x __attribute__((unused))

```

-----8<-----

If you prefer git pull:

git pull git://80x24.org/ruby.git gc-guard-harder-v2

Maybe my original idea works, too, but it is weaker, I think:

git pull git://80x24.org/ruby.git gc-guard-harder

<http://bogomips.org/ruby.git/patch?id=ecc6f50ca>

#18 - 02/20/2014 11:10 PM - nobu (Nobuyoshi Nakada)

Seems better.

Let's try.

#19 - 02/20/2014 11:46 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset [r45064](#).

gc.c: RB_GC_GUARD should be robust enough for any compiler

- include/ruby/ruby.h (RB_GC_GUARD): use rb_gc_guarded_ptr_val on non-GCC/MSVC
- gc.c (rb_gc_guarded_ptr_val): rename and adjust argument. RB_GC_GUARD should be robust enough for any compiler. [Bug [#7805](#)]

#20 - 02/20/2014 11:53 PM - normalperson (Eric Wong)

Thanks for taking a look, [r45064](#)

Files

marshal-c-volatile.patch	591 Bytes	02/22/2013	ngoto (Naohisa Goto)
--------------------------	-----------	------------	----------------------