Ruby master - Bug #16807
Ruby 2.7 segfault loading sources under GC.stress
04/22/2020 08:03 AM - qnighy (Masaki Hara)

| Status:          | Closed         |
| Priority:        | Normal         |
| Assignee:        | qnighy         |
| Target version:  | ruby 2.7.1p83  |

Backport: 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONE

Description
Originally seen in https://github.com/protocolbuffers/protobuf/pull/7386, but reproduces without C extension.

Reproduction: place these four files from this gist:

- tests/
  - gc_test.rb
  - generated_code_pb.rb
  - generated_code_proto2_pb.rb
- lib/
  - google/
    - protobuf.rb

Then I got the following:

$ ruby -Ilib tests/gc_test.rb
/Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72: [BUG] Segmentation fault at 0x00007fd23ba8e608
ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [x86_64-darwin19]

-- Crash Report log information -----------------------------------------------
See Crash Report log file under the one of following:
  * ~/Library/Logs/DiagnosticReports
  * /Library/Logs/DiagnosticReports
for more details.
Don't forget to include the above Crash Report log file in bug reports.

-- Control frame information --------------------------------------------------
c:0007 p:----- s:00052 e:000051 CFUNC :require
   at /Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72: in `require'
c:0006 p:0111 s:0047 e:000046 METHOD /Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72
   at /Users/qnighy/workdir/ruby-parse-failure/tests/generated_code_pb.rb:4: in `require'
c:0004 p:----- s:0028 e:000027 CFUNC :require
   at /Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72: in `require'
c:0003 p:0111 s:0023 e:000022 METHOD /Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72
   at /Users/qnighy/workdir/ruby-parse-failure/tests/generated_code_pb.rb:4: in `require'
c:0002 p:0064 s:0007 E:000038 EVAL tests/gc_test.rb:8 [FINISH]
c:0001 p:0000 s:0003 E:001d80 (none) [FINISH]

-- Ruby level backtrace information -------------------------------------------
tests/gc_test.rb:8:in `<main>'
   at /Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72:in `require'
/Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72:in `require'
/Users/qnighy/workdir/ruby-parse-failure/tests/generated_code_pb.rb:4:in `<top (required)>'
/Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72:in `require'
/Users/qnighy/.rbenv/versions/2.7.1/lib/ruby/2.7.0/rubygems/core_ext/kernel_require.rb:72:in `require'

-- Machine register context --------------------------------------------------
rax: 0x00000000c27c19ab rbx: 0x0000000000000000 rcx: 0x00000000c27c818b0
rdx: 0x00000000c27c19ab rdi: 0x0000000000000000 rsi: 0x00000000c27c19ab rbp: 0x0000000000000000 rsp: 0x00007fcc27c0b030 r8: 0x0000000000000000 r9: 0x0000000000000028 r10: 0x00007fcc27c0b030 r11: 0xffffffcd3acd8990 r12: 0x0000000000000000 r13: 0x0000000000000000 r14: 0x0000000000000000 r15: 0x0000000000000000 rip: 0x0000000102a3724e rfl: 0x0000000000010282

-- C level backtrace information -------------------------------------------
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_print_backtrace+0x19) [0x102be8a09]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_vm_bugreport+0xa8) [0x102be8ad8]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_bug_for_fatal_signal+0x1b5) [0x10296c4485]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(sig_do_nothing+0x0) [0x102b11540]
/usr/lib/system/libsystem_platform.dylib(_sigtramp+0x1d) [0x7fff6a8435fd]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(mark_ast_value+0x6e) [0x102a3724e]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(iterate_buffer_elements+0x51) [0x102a37d81]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(iterate_node_values+0x38) [0x102a36fb8]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_ast_mark+0x81) [0x102a371d1]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_mark_imemo+0x206) [0x1029a5ce6]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_mark_children+0x123) [0x102997173]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_mark_stacked_objects+0x79) [0x10299fe69]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_mark_stacked_objects_all+0x1b) [0x1029a0a4b]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_marks_rest+0x7f) [0x1029a18e7]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_marks+0x67) [0x1029a5ce6]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(gc_start+0x364) [0x10299d944]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(garbage_collect+0x24) [0x1029947d4]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(garbage_collect_with_gvl+0x5f) [0x1029a63bf]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(objspace_malloc_gc_stress+0x8a) [0x1029a635a]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(objspace_xmalloc0+0x2a) [0x1029a790e4]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_parser_compile_file_path+0x7e) [0x1029a811ee]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(load_file+0x6c) [0x102b0813c]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_parser_load_file+0x83) [0x102b07dd3]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(require_internal+0x505) [0x1029eca35]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(rb_require_string+0x27) [0x1029ebbb7]
/Users/qnighy/.rbenv/versions/2.7.1/lib/libruby.2.7.dylib(yylex+0x34) [0x102a790e4]
Don't forget to include the Crash Report log file under DiagnosticReports directory in bug reports.

zsh: abort      ruby -Ilib tests/gc_test.rb
This is seen in Linux too in the aforementioned pull request, but I couldn't reproduce it in my local Linux machine.

Also reproduced with miniruby from the latest master (8474e6a).

**Associated revisions**

**Revision b35bfa6 - 10/01/2020 12:02 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) 35ba2783fe6b3316a6bb6c6f00bf975ad7185d6e0,e8edc34f0abeb176b24975a1fed12c37820a252: [Backport #16807]

Use a linked list to eliminate imemo tmp bufs for managing local tables

This patch changes local table memory to be managed by a linked list rather than via the garbage collector. It reduces allocations from the GC and also fixes a use-after-free bug in the concurrent-with-sweep compactor I'm working on.

Remove unused struct member

I accidentally added this in 35ba2783fe6b3316a6bb6c6f00bf975ad7185d6e0, and it's making the size of RVALUE be too big. I'm sorry! orz

**History**

#1 - 04/22/2020 09:49 AM - nobu (Nobuyoshi Nakada)

ID list in NODE_SCOPE seems corrupted.
Maybe a GC compaction related issue?

#2 - 10/01/2020 11:39 AM - mame (Yusuke Endoh)

This issue still reproduces on ruby_2_7 branch, and does not on master.

I investigated the issue. local_tbl() allocates imemo_tmpbuf for a local variable table and returns a pointer to the buffer. And then, node_newnode_withlocals creates an AST node with the pointer. However, if GC may occur on the allocation of the new NODE, the table is freed, so the NODE contains a dangling pointer, which cause a very rare segfault issue.

I created a patch to fix the issue. But it does not apply to master because the relevant functions have been drastically refactored at 35ba2783fe6b3316a6bb6c6f00bf975ad7185d6e0, which also (maybe unknowingly) fixes the issue. So, we may backport the commit to ruby_2_7.

```
diff --git a/parse.y b/parse.y
index dc9c37555..7734beecdf 100644
--- a/parse.y
+++ b/parse.y
@@ -518,7 +518,7 @@ static NODE *symbol_append(struct parser_params *p, NODE *symbols, NODE *symbol)
 static NODE *match_op(struct parser_params*,NODE*,NODE*,const YYLTYPE*,const YYLTYPE*);
 -static ID  *local_tbl(struct parser_params*);
 +static VALUE local_tbl(struct parser_params*);
 +static VALUE local_tbl(struct parser_params*);
 +static VALUE reg_compile(struct parser_params*, VALUE, int);
 static void reg_fragment_setenc(struct parser_params*, VALUE, int);
 @@ -11819,7 +11819,7 @@ local_pop(struct parser_params *p)
 }
 #ifndef RIPPER
-#static ID*
+static VALUE
 local_tbl(struct parser_params *p)
 {
   int cnt_args = vtable_size(p->lvtbl->args);
@@ -11849,17 +11849,17 @@ local_tbl(struct parser_params *p)
   buf[cnt + 1] = (ID)tbl;
   RB_OBJ_WRITTEN(p->ast, Qnil, tbl);
   return buf;
 + return tbl;
 }
 static NODE *
 node_newnode_with_locals(struct parser_params *p, enum node_type type, VALUE a1, VALUE a2, const rb_code_location_t *loc)
 {
-  ID *a0;
-  NODE *n;
-  a0 = local_tbl(p);
+  VALUE tbl = local_tbl(p);
+  ID *a0 = tbl ? RB_IMEMO_TMPBUF_PTR(tbl) : 0;
    n = NEW_NODE(type, a0, a1, a2, loc);
    + RB_GC_GUARD(tbl);
```
def report
    return n;
}

#3 - 10/01/2020 11:42 AM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED
- Status changed from Open to Closed

Thank you for your report.

It seems that the git:35ba2783fe6b3316a6bbcf0f0bf975ad7185d6e0 fixed this issue.
I will backport it to ruby_2_7 branch.

#4 - 10/01/2020 12:02 PM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONE

ruby_2_7 b35bfa6abb7760e4323a4341df840f59ddcfde1 merged revision(s)
35ba2783fe6b3316a6bbcf0f0bf975ad7185d6e0,e8edc34f0abe176b24975a11ed1f2c3782f0a252.